Tampereen yliopisto

Tuuli Vattulainen

TILAAMISEN TODENNÄKÖISYYDEN ENNUSTAMINEN VERKKOKÄYTTÄYMISESTÄ KONEOPPIMISMENETELMIN

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Helmikuu 2020

TIIVISTELMÄ

Tuuli Vattulainen: Tilaamisen todennäköisyyden ennustaminen verkkokäyttäytymisestä koneoppimismenetelmin
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Helmikuu 2020

Palvelun tilaajakannan kasvattamiseen kuuluu kaksi peruselementtiä: uusien tilaajien hankinta ja vanhojen tilaajien pito. Tässä tutkielmassa keskitytään uusien asiakkaiden hankintaan. Tutkimusongelmana on palvelun tilaamisen todennäköisyyden ennustaminen verkkokäyttäytymisen perusteella. Käytetyt algoritmit ovat todennäköisyyksien ennustamiseen soveltuvat päätöspuut C4.5 ja CART sekä satunnaismetsä.

Tässä kontekstissa tilaaminen on melko harvinainen tapaus, ja käsiteltävä datan luokkajakauma on epätasapainossa. Tutkielmassa keskitytään epätasapainoisen datan käsittelytapoihin. Otantamenetelmistä testattavaksi on valittu satunnainen aliotanta ja SMOTE, ja lisäksi otantamenetelmiä kokeillaan viidellä luokkasuhteella. Satunnaisessa aliotannassa datasta poistetaan satunnaisesti enemmistöluokan havaintoja ja SMOTEssa yhdistettynä aliotantaan sekä poistetaan enemmistöluokan havaintoja että luodaan vähemmistöluokan havaintojen pohjalta uusia synteettisiä havaintoja. Koska otanta vaikuttaa ennustettaviin todennäköisyyksiin, ennustettujen todennäköisyyksien kalibroinnissa hyödynnetään Plattin skaalausta tai isotonista regressiota.

Luokittelualgoritmeja, otantamenetelmiä, otantasuhteita sekä kalibrointimenetelmiä yhdistelemällä luotiin yhteensä 90 erilaista ennustemallia. Malleja verrattiin logaritmisen tappion sekä Brierin pisteiden avulla, jotka ovat todennäköisyyksien ennustamisessa yleisesti käytettyjä evaluointimetriikoita. Parhaaksi ennustajaksi osoittautui malli, jossa yhdistettiin satunnaismetsä, SMOTE luokkasuhteella 4:1 ja isotoninen regressio. SMOTE toimi satunnaista aliotantaa paremmin ja isotoninen regressio toimi SMOTEn kanssa Plattin skaalausta paremmin.

Testauksessa parhaan mallin logaritminen tappio on 0,1 ja Brierin pisteet 0,02. Kun ennustetut todennäköisyydet binärisoidaan luokiksi raja-arvolla t = 0,02, saadaan mallin tarkkuudeksi, sensitiivisyydeksi ja spesifisyydeksi 0,6. Rajaa pienentämällä saadaan sensitiivisyyttä kasvatettua väärien positiivisten kustannuksella. Tarvittaessa siis tunnistetaan hyvin tilanneita, mutta silloin malli ennustaa myös paljon ei-tilanneita tilanneiksi. Luokittelukyky jää kokonaisuudessaan melko heikoksi, mutta toisaalta tutkimusongelman kannalta tärkeintä on havaita potentiaaliset tilaajat, joten väärät positiiviset eivät ole välttämättä kovin haitallisia.

Avainsanat: todennäköisyyden ennustaminen, epätasapainoinen data, C4.5, CART, satunnaismetsä, otantamenetelmät, kalibrointi

Sisällysluettelo

1 Johdanto

Pro gradu -tutkielman toimeksiantaja on suomalainen monialayritys, joka työllistää tällä hetkellä yli 500 ihmistä Suomessa ja kansainvälisesti. Yrityksen liikevaihto on yli 100 miljoonaa euroa. Yrityksen yhtenä liiketoiminnan osa-alueena on erään tuotteen tilaajakannan kasvattaminen, johon liittyy kaksi peruselementtiä: uusien tilaajien hankinta ja asiakaspito. Tässä tutkielmassa keskitytään uusien tilaajien hankintaan ja selvitetään koneoppimisen avulla, voiko verkkokäyttäytymisestä ennustaa verkkovierailijan halukkuutta ryhtyä yrityksen tuotteen tilaajaksi.

Täsmällisenä tutkimusongelmana on selvittää, voiko tilaamisen todennäköisyyttä ennustaa verkkokäyttäytymisen perusteella ja kuinka tarkasti. Sovellusalueella tilaaminen on harvinainen tapahtuma. Tutkitun datan luokkajakauma on siksi hyvin epätasapainossa: tilanneita on vain 1,8 % kaikista havainnoista. Tutkielman keskeisenä tavoitteena on tutkia, miten tällaista epätasapainoista dataa käsitellään, sillä yleisesti ottaen tämä on ongelma koneoppimisalgoritmeille. Yksi syy tälle on, että useat luokittelualgoritmit hyödyntävät tarkkuutta mallia rakentaessa eli yrittävät minimoida virhettä [Visa and Ralescu. 2005]. Tutkielmassa ongelmaa ratkotaan otantamenetelmien, kalibroinnin ja mallin evaluointimetriikoiden avulla. Otantamenetelmistä testataan satunnaisotantaa sekä SMOTEa aliotannalla [Chawla, et al. 2002].

Koska tutkielmassa ennustetaan tapauksen todennäköisyyksiä kuulua luokkaan, tarkasteltaviksi algoritmeiksi on valittu jo luonnostaan todennäköisyysperusteisia menetelmiä: päätöspuualgoritmi C4.5 [Quinlan. 1993], päätöspuualgoritmi CART [Breiman, et al. 1984] ja satunnaismetsä [Breiman. 2001]. Näiden antamat ennusteet kalibroidaan vielä vastaamaan paremmin todellisia todennäköisyyksiä. Tutkielmassa testataan sekä Plattin skaalausta [Platt. 1999] että isotonista regressiota [Robertson, et al. 1988].

Luvussa 2 kuvaillaan tarkemmin sovellusongelman luonnetta ja tarkastellaan dataa. Luvussa 3 esitellään tutkielmassa käytettyjä menetelmiä, joista ensiksi esitellään menetelmiä epätasapainoisen luokkajakauman hallintaan (luku 3.1). Luvussa 3.2 esitellään käytetyt luokittelualgoritmit, luvussa 3.3 mallien evaluoinneissa käytetyt metriikat logaritminen tappio, Brierin pisteet ja sekaannusmatriisi ja luvussa 3.4 kalibrointimenetelmät. Luvussa 4 kuvaillaan testiasetelmat ja luvussa 5 tarkastellaan saatuja tuloksia. Loppuyhteenveto ja jatkoehdotuksia käsitellään luvussa 6.

2 Sovellusongelman ja datan kuvaukset

2.1 Tutkittava ongelma

Tilaajakannan kasvattamisen kannalta on oleellista asiakaspidon lisäksi saada uusia tilauksia. Tässä tutkielmassa tutkitaan, voiko verkkokäyttäytymisen perusteella päätellä kävijän tilaamistodennäköisyyttä ja kuinka tarkkoihin ennusteisiin päästään. Yksi esimerkki tilaamistodennäköisyyden hyödyntämisestä on kohdentaminen sivustolla. Jos havaitaan, että sivustolla vierailevan kävijän tilaamisen todennäköisyys ylittää tietyn rajan, hänelle voidaan kohdentaa markkinointia, joka tukisi tilaamista. Toisaalta markkinointia ei kannata kohdentaa kaikille, koska sivustolla on rajallinen määrä tilaa: sellaisille kävijöille, jotka eivät todennäköisesti aio tilata, on liiketoimintamielessä arvokkaampaa näyttää jotain muuta. Tässä tutkielmassa ei keskitytä tunnistamisen jälkeisiin toimenpiteisiin, vaan tutkitaan, miten koneoppimista voidaan hyödyntää potentiaalisten asiakkaiden tunnistamiseen.

Koneoppimisen tehtävät voidaan jakaa ohjattuun ja ohjaamattomaan oppimiseen [Han and Kamber. 2001]. Klusterointi on esimerkki ohjaamattomasta oppimisesta, jossa havaintoja niputetaan samaan joukkoon niiden samankaltaisuuden perusteella. Niiden luokkaa tai tyyppiä ei siis tiedetä etukäteen, vaan kuvailuja tehdään muodostuneiden klusterien perusteella. Ohjatussa oppimisessa on puolestaan jokin ennustettava muuttuja. Kategorisen muuttujan tapauksessa voidaan puhua luokista ja luokittelu onkin esimerkki ohjatusta oppimisesta. Tätä sovellusongelmaa ratkotaan nimenomaan luokittelun avulla ja luokkatietona on se, onko kävijä tilannut vai ei.

Luokittelun idea on luoda funktio *f*, jolla havainto X kuvataan luokkaan y. Tyypillisesti luokittelijan mallinnuksessa käytettävissä oleva data jaetaan harjoitusdataan ja testidataan, esimerkiksi suhteessa 9:1 tai 8:2. Harjoitusdata sisältää havainnot X_i, havaintoja kuvailevat muuttujat A_i ja luokkamuuttujan C, jota halutaan ennustaa. Algoritmi yrittää ratkaista, millä havaintojen muuttuja–arvo-yhdistelmillä voidaan ennustaa eri luokkia parhaiten. Tämän jälkeen koulutettua mallia evaluoidaan sivuun jätetyllä testidatalla, jota malli ei ole hyödyntänyt koulutuksessa. Koulutetulle mallille syötetään testidata lukuun ottamatta luokkamuuttujaa C. Ennustamisen jälkeen voidaan verrata mallin tuottamia ennusteita ja muuttujan C arvoja eli tunnettuja oikeita luokkia. Kun malli on todettu riittävän hyväksi, voidaan uusia havaintoja luokitella syöttämällä havainnon muuttujan arvot funktiolle *f*, joka siten palauttaa ennustetun luokan. Tyypillisiä luokitteluun käytettäviä algoritmeja ovat muun muassa lineaariset luokittelijat, tukivektorikoneet, päätöspuut ja satunnaismetsät, neuroverkot sekä *k*:n lähimmän naapurin luokittelualgoritmi.

Tilaamisen todennäköisyyden ennustamiseen liittyy joitakin ennalta tiedettyjä haasteita, esimerkiksi se, että tilaaminen on harvinainen tapahtuma. Kerätyssä datassa vain 1,8 % havainnoista on tilanneita kävijöitä. Näin suuri epätasapaino datan luokkajakaumassa on tyypillisesti haastavaa luokittelualgoritmeille. Tätä ratkotaan käyttämällä sopivia metodeja luokkajakaumaltaan vinon datan hallintaan.

Kävijän yksilöinti perustuu verkkoselaimen evästeeseen, johon liittyy yksilöivä tunnistetieto. Haasteen tuo se, että yksi ihminen voi käyttää useaa verkkoselainta esimerkiksi pöytätietokoneelta ja kännykältä ja vastaavasti yhtä verkkoselainta voi käyttää useampi ihminen, esimerkiksi perhe yhteiseltä tietokoneeltaan. Toiseksi, mikäli verkkoselaimen evästeet poistetaan, poistuu myös sinne asetetut tunnistetiedot, joiden avulla kävijä yksilöidään datassa. Tällöin seuraavan kerran saapuessaan kävijä näyttäytyy mittauksessa uutena kävijänä. Jälkimmäistä ongelmaa on ratkottu siten, että dataan on rajattu vain sellaiset kävijät, joilla tiedetään olevan vähintään 14 päivän historia, vaikka tällöin kaikkia tilanneita ei saada mukaan dataan.

Verkkokäyttäytymisestä kerätään dataa Google Analytics -työkalun avulla ja se on tallennettu tietokantaan, josta tutkielmassa käytetty data on muodostettu SQL-kyselyin. Alkuperäinen data on istuntotasoista. Istunto tarkoittaa yhtä vierailua sivustolla: istunto alkaa, kun kävijä saapuu palveluun ja päättyy, kun kävijä poistuu palvelusta. Istunnosta on kerätty muun muassa tulotapaan, laitteeseen, verkkoselaimeen ja aikaan liittyviä tietoja. Yksi istunto puolestaan koostuu yhdestä tai useammasta sivulatauksesta. Jokaisesta sivulatauksesta on kerätty sivuun liittyviä tietoja. Sivuun liittyviä tietoja on muun muassa sivun aihe, kirjoittaja tai sivulatauksen aika. Istunto voi sisältää myös niin kutsuttuja tapahtumia. Esimerkiksi sivun vierittäminen alaspäin tai kommentin jättäminen ovat tapahtumia.

Data on koostettu havainnoiksi siten, että yksi havainto käsittää yhden kävijän tiedot 14 päivän ajalta. Tilanneiden osalta dataa haetaan tilauspäivästä 14 päivää taaksepäin tilauspäivä pois lukien. Siis tilauspäivän dataa ei huomioida. Nämä muodostavat positiivisen luokan ”tilanneet”. Kustannusten säästämiseksi muiden kuin tilanneiden osalta on valittu joka 20. päivä, josta taaksepäin historiadataa haetaan vastaavasti kuin tilanneille. Nämä havainnot muodostavat negatiivisen luokan, ”ei-tilanneet”. Näin on muodostettu 14 päivän tarkasteluvälit sekä tilanneille että ei-tilanneille siten, että tilauspäivät eivät sisälly dataan.

Alkuperäisessä datassa on 338 062 havaintoa eli verkkosivustolla vieraillutta kävijää ja 203 muuttujaa, joita on sekä numeerisia että kategorisia. Tilanneita kävijöitä on 5 977 ja

muita 332 085, eli positiivisia havaintoja on 1,8 %. Esikäsittelyn aikana turhia muuttujia, kuten tilauspäivä- ja tunnistemuuttujat, poistettiin ja vahvasti korreloivista (korrelaatiokerroin yli 0,75) muuttujista poistettiin se, joka korreloi eniten muiden muuttujien kanssa. Korrelaatioanalyysin perusteella karsittiin 35 muuttujaa. Kategoriset muuttujat muutettiin dummy-muuttujiksi, jolloin nekin voitiin huomioida korrelaatiokertoimien laskuissa. Muita muuttujan valintametodeja ei käytetty, sillä luokittelualgoritmit itsessään valitsevat tärkeimmät muuttujat malliin.

2.2 Muuttujat

Muuttujien arvojen vaihtelut luokittain vaikuttavat verrattain pieniltä, vaikka eroja löytyykin. Tämä antaa viitteitä siitä, ettei ennustaminenkaan välttämättä ole helppoa. Lisäksi täytyy muistaa datan osin epätarkka luonne eli se, että todellisen ihmisen yksilöinti on haastavaa: yksi datan havainto voi vastata useaa eri ihmistä ja toisaalta yksi ihminen voi olla datassa eri havaintoina. Tämä varmasti osaltaan heikentää ennustettavuutta. Seuraavaksi esitellään tarkemmin yksittäisiä muuttujia, jotka ovat lopullisessa valitussa mallissa tärkeimpiä muuttujia. Muuttujien tärkeys on päätelty MDA-metriikalla (*engl. Mean Decrease Accuracy*), joka esitellään tarkemmin luvussa 3.2.4.

Käyttäytymiseen liittyvät muuttujat ovat laskettu pääsääntöisesti koko tarkasteluväliltä. Esimerkiksi käyntien tai sivulatausten lukumäärät tarkoittavat tarkasteluvälin aikana tehtyjä käyntejä ja sivulatauksia.

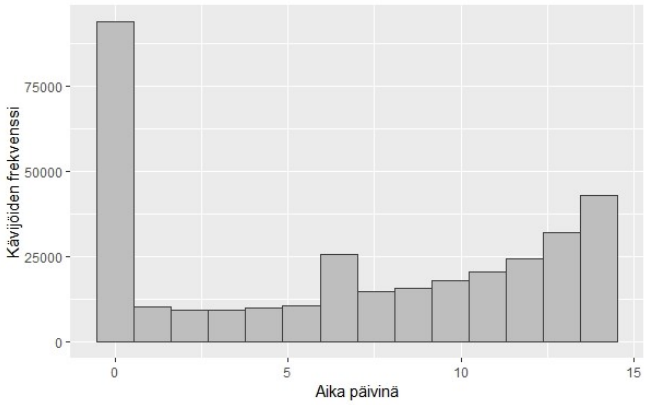
Ostoputkessa käynti tilaamatta
Tilausta tehdessä kävijä siirtyy niin sanottuun ostoputkeen. Putken ensimmäisessä vaiheessa valitaan tuote, toisessa luodaan tunnus, kolmannessa annetaan yhteystiedot ja neljännessä vaiheessa valitaan maksutapa ennen varsinaista maksutapahtumaa. Keskeytyneen tilauksen teon voidaan ajatella ilmentävän ostohalukkuutta ja on siten luonnollista, että tämä muuttuja nousi tärkeimpien muuttujien joukkoon tilanneita ja ei-tilanneita erottelevaksi muuttujaksi. Tilanneet ovat vierailleet tarkasteluvälin aikana ostoputkessa useammin, mutta tilauksen teko on jäänyt kesken sillä kerralla (taulukko 1).

	Positiivinen	**Negatiivinen**
Käyty ostoputkessa = ei	94 % (5622)	99 % (327599)
Käyty ostoputkessa = kyllä	6 % (355)	1 % (4486)

Taulukko 1: Ostoputkessa käynti luokittain

Päivien lukumäärä tarkasteluvälin ensimmäisen ja viimeisen käynnin välillä

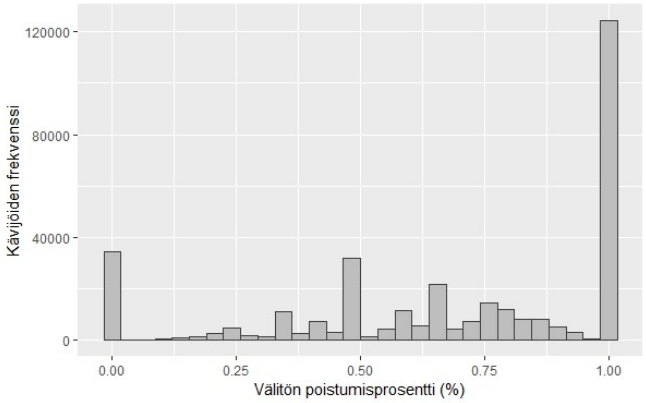
Tässä viimeisellä käynnillä tarkoitetaan viimeistä istuntoa tarkasteluvälillä. Ensimmäinen ja viimeinen käynti voi olla myös sama istunto, jolloin istuntojen välisten päivien lukumäärä on 0. Valtaosa nollista on tällaisia istuntoja. Kokonaisuudessaan jakauma on nähtävillä histogrammissa alla (kuva 1). Luokittain tarkasteltuna negatiivisella luokalla keskiarvo on 6,8 ja mediaani on 8 ja positiivisella luokalla keskiarvo on 5,9 ja mediaani 6.



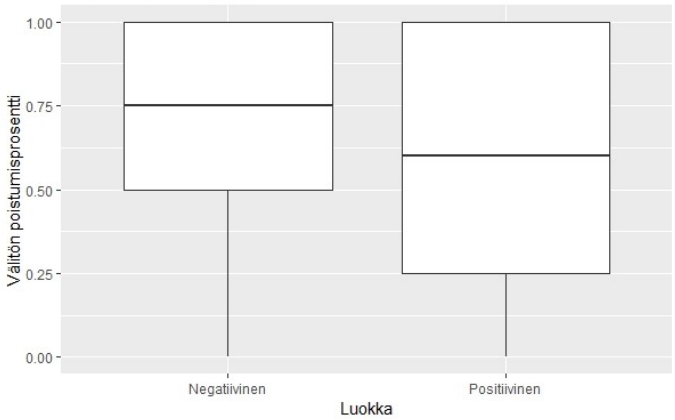
Kuva 1: Histogrammi päivien lukumäärästä ensimmäisen ja viimeisen käynnin välillä

Välitön poistumisprosentti

Välitön poistuminen tarkoittaa sellaista istuntoa, jossa kävijä poistuu samalta sivulta, jolta istunto alkoi eikä siten siirry verkkosivustolla muille sivuille. Välitön poistumisprosentti tarkoittaa välittömien poistumien osuutta kaikista kävijän istunnoista tarkasteluvälillä. Kuvan 2 histogrammista huomataan piikki arvon 1 kohdalla: aineiston kävijöistä 37 % on sellaisia, joiden kaikki istunnot ovat myös päättyneet aloitussivulleen. Kuvasta 3 nähdään, että negatiivisella luokalla välitön poistumisprosentti on yleisesti ottaen suurempaa (mediaani 0,75) kuin positiivisella luokalla (mediaani 0,60). Jakauma vastaa tilaajien ja ei-tilaajien oletettua käyttäytymistä, sillä oletuksena on, että tilaajat kuluttavat palvelua enemmän ja siten tekevät palvelussa siirtymiä.



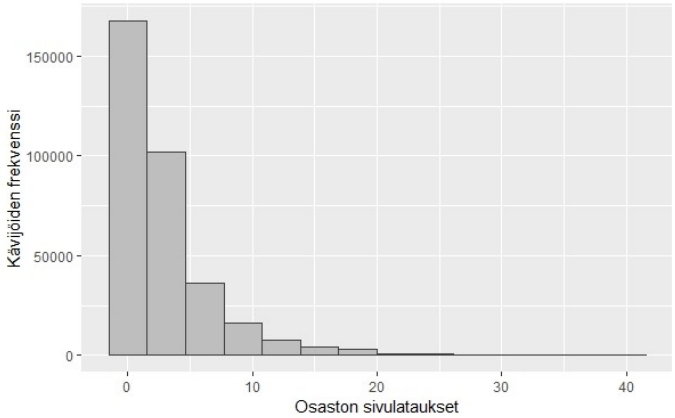
Kuva 2: Välitön poistumisprosentti



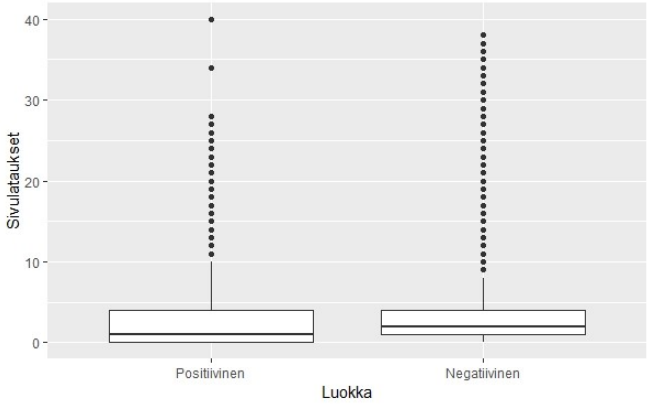
Kuva 3: Välitön poistumisprosentti luokittain

Tietyn osaston sivulataukset

Tietyn sivuston osaston, joka on tässä anonymisoitu, sivulatausten mediaani on 2 ja keskiarvo 2,8 (kuva 4). Luokittain tarkasteltuna jakaumat ovat lähellä toisiaan, mutta negatiivisen luokan mediaani on hiukan suurempi (kuva 5).



Kuva 4: Osaston sivulataukset

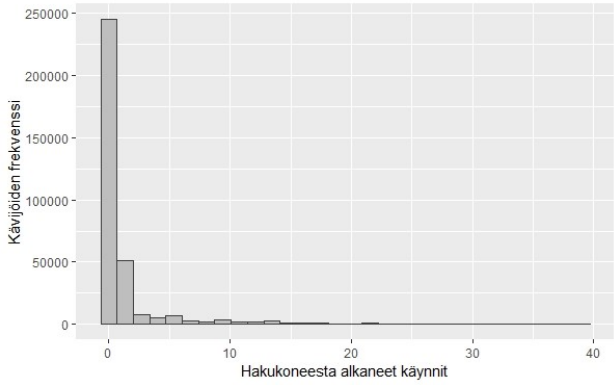


Kuva 5: Osaston sivulataukset luokittain

Hakukoneista alkaneiden istuntojen lukumäärä

Dataan on kerätty tietoa istunnon tulotavoista. Tulotapa voi olla esimerkiksi sosiaalinen media, hakukone tai suoraan sivustolle saapuminen. Mallinnuksesta havaittiin, että hakukoneista tulleiden istuntojen yhteenlaskettu lukumäärä tarkasteluvälillä on tärkeä muuttuja mallissa.

Histogrammissa (kuva 6) on hakukoneista alkaneiden istuntojen lukumäärän jakauma. Valtaosa kävijöistä ei ole kertaakaan tullut minkään hakukoneen kautta. Luokittaisessa tarkastelussa jakaumissa ei näy suurta eroa, mutta keskiarvo on hiukan suurempi negatiivisella luokalla (1,31) positiivisen luokan keskiarvon ollessa 1,22. Koska hakukoneista tulee liikennettä suurimmaksi osaksi brändiin liittyvillä hakusanoilla, voisi suuri hakukoneista alkaneiden istuntojen lukumäärä kieliä sitoutuneisuudesta palveluun. Näin ollen olisi voinut olettaa istuntojen lukumäärän olevan suurempi tilaajilla kuin ei-tilaajilla.

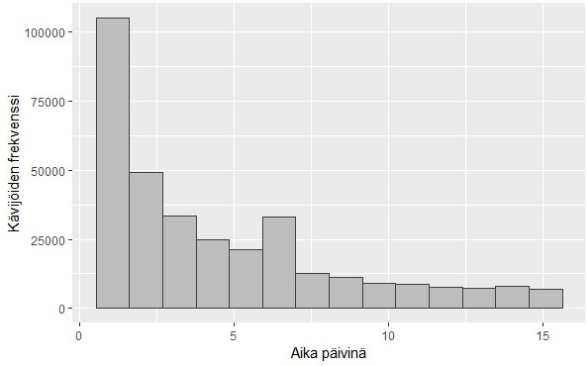


Kuva 6: Hakukoneista tulleiden käyntien lukumäärä

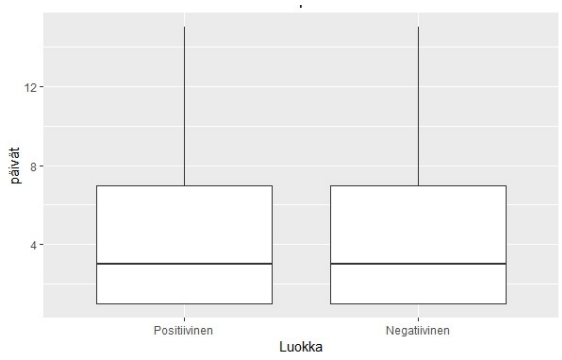
Edellisestä käynnistä kuluneiden päivien lukumäärä

Mallissa yksi tärkeä muuttuja on tilausistunnon tai tarkastelupäivän ja sitä edeltävän käynnin välissä olevien päivien lukumäärä. Tilanneiden tapauksessa tarkoitetaan siis,

kauanko tilauspäivänä on edellisestä käynnistä kulunut päiviä. Jakaumasta (kuva 7) nähdään, että iso osa on käynyt päivä tai pari ennen tilausta tai tarkastelupäivää myös sivustolla. Jos tarkastellaan laatikko–jana-kuvioita luokittain, on lähes mahdotonta havaita luokkakohtaisia eroja (kuva 8).



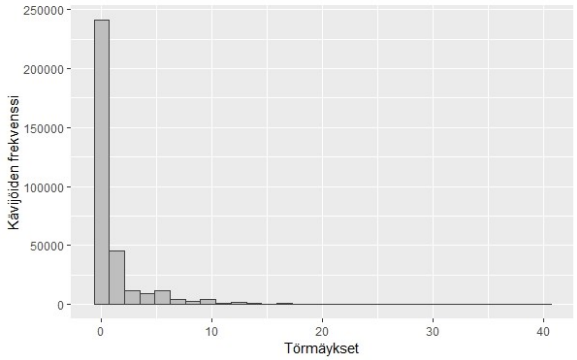
Kuva 7: Edellisestä käynnistä kuluneiden päivien lukumäärä



Kuva 8: Edellisestä istunnosta kuluneiden päivien lukumäärä luokittain

Mittaroidun muurin törmäykset tarkasteluvälillä

Sivustolla osaa sisältöä voi kuluttaa vain tietyn verran viikossa ilman tilausta, kunnes sisältö menee lukkoon. Tällaisiin tapahtumiin viitataan termillä mittaroidun muurin törmäys. Törmäysten jakauma näkyy kuvassa 9 ja jakaumat luokittain taulukossa 2. Ei-tilanneet kävijät ovat törmänneet lukumäärällisesti enemmän mittaroituun muuriin kuin tilanneet.



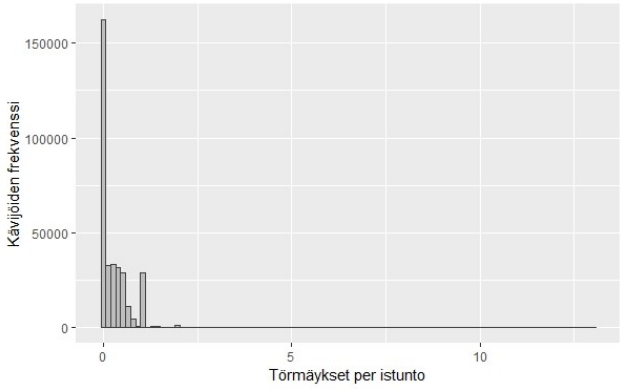
Kuva 9: Mittaroidun muurin törmäykset tarkasteluvälillä

	Min	25 %	50 %	75 %	Max	Ka
Positiivinen	0	0	0	0	40	0,6
Negatiivinen	0	0	0	1	34	1,2

Taulukko 2: Mittaroidun muurin törmäykset tarkasteluvälillä luokittain

Kovan maksumuurin törmäysten keskimääräinen lukumäärä istunnossa

Osa sisällöstä on tarkoitettu vain tilaajille. Tällainen sisältö on niin kutsutun kovan maksumuurin takana, johon tilaukseton kävijä törmää sisällön sivulle tullessaan. Tämä muuttuja koostuu käyttäjän tarkasteluvälin kovan maksumuurin törmäyksistä istuntojen lukumäärällä jaettuna. Mittaroidun muurin törmäyksistä poiketen, kovan muurin törmäysten lukumäärä istunnossa on yleisesti suurempi tilanneilla kuin ei-tilanneilla. Kuvasta 10 nähdään yleinen jakauma ja taulukosta 3 jakaumat luokittain. Positiivisella luokalla mediaani on 0,2 ja negatiivisella 0,09.



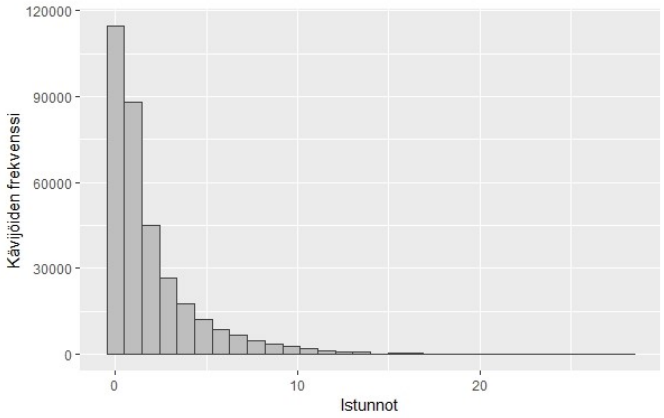
Kuva 10: Kovan muurin törmäykset istunnossa

	Min	25 %	50 %	75 %	Max	Ka
Positiivinen	0	0	0,2	0,5	5	0,33
Negatiivinen	0	0	0,09	0,4	13	0,25

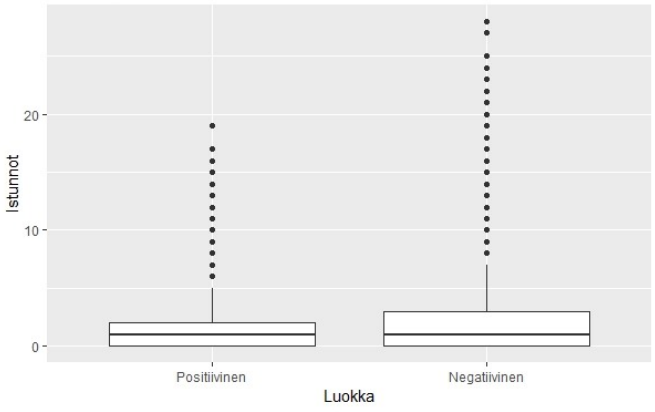
Taulukko 3: Kovan muurin törmäykset istunnossa luokittain

Illalla tehdyt istunnot

Tässä tutkielmassa illalla tarkoitetaan kellonaikaa 17–23. Kuvassa 11 esitellään yleinen jakauma iltaistuntojen lukumäärälle tarkasteluvälillä ja kuvassa 12 on tarkastelu luokittain. Negatiivisilla havainnoilla on enemmän istuntoja iltaisin. Negatiivisen luokan keskiarvo on 2,01, kun taas positiivisella luokalla se on 1,72.



Kuva 11: Iltaistuntojen lukumäärä tarkasteluvälillä



Kuva 12: Istuntojen lukumäärä illalla luokittain

3 Menetelmät

3.1 Datan luokkajakauman epätasapainon hallinta

Tutkittavan datan luokkajakauma on selvästi epätasapainossa, sillä positiivisia havaintoja on vain 1,8 % kaikista havainnoista. Todellisuudessa osuus on vieläkin pienempi, sillä haettaessa negatiivisia havaintoja tietokannasta tehtiin jo satunnaisotantaa kustannussyiden takia. Tässä luvussa esitellään lyhyesti kirjallisuudessa esitettyjä keinoja luokkajakaumaltaan vinon datan käsittelyyn ja tarkemmin tässä tutkielmassa käytetyt metodit.

Monissa tutkimuksissa on todettu, että luokkajakaumaltaan epätasapainoinen data aiheuttaa ongelmia perinteisille luokittelualgoritmeille, ja saa ne painottamaan enemmistöluokkaa tuloksissaan. Kolmeksi tärkeimmäksi syyksi esitetään [Visa and Ralescu. 2005]:

1. Luokittelualgoritmien evaluoinnissa hyödynnetään tarkkuutta eli yritetään minimoida luokittelun virhe, johon vähemmistöluokka vaikuttaa vain vähän pienuutensa vuoksi. Esimerkiksi tämän tutkielman aineistossa, tarkkuudeksi saadaan 0,982, mikäli kaikki testiaineiston havainnot määritetään suoraan enemmistöluokkaan.
2. Algoritmit olettavat, että käsiteltävä data on jakaumaltaan samanlainen kuin harjoitusaineistossa.
3. Algoritmit olettavat lisäksi, että eri luokkavirheillä on samanlainen painoarvo.

Seuraavaksi esitellään yleiset metodit luokkajakaumaltaan epätasapainoisen datan käsittelyyn. Teoriakokonaisuus perustuu suurelta osin kokonaisuudessaan katsausartikkeliin [Kotsiantis, et al. 2006]. Metodit epätasapainoisen datan käsittelyyn voidaan jakaa alakategorioihin: datatason menetelmiin, algoritmitason menetelmiin, näiden yhdistelmiin sekä evaluointimetriikoiden käyttöön.

3.1.1 Datatason menetelmät

Datatason menetelmiin kuuluvat eri otantamenetelmät ja sopivat muuttujan valintametriikat. Otantamenetelmät jakaantuvat vielä kahtia sen perusteella, karsitaanko enemmistöluokan havaintoja vai kasvatetaanko havaintojen lukumäärää vähemmistöluokassa. Aliotantaan (*engl. undersampling)* kuuluu esimerkiksi satunnainen aliotanta, missä enemmistöluokasta poistetaan satunnaisesti havaintoja [Kotsiantis, et al. 2006]. Myös klusterointiin perustuva otantamenetelmä kuuluu aliotantamenetelmiin. Siinä enemmistöluokka klusteroidaan n klusteriin, missä n on vähemmistöluokan

havaintojen lukumäärä. Sen jälkeen jokaisesta klusterista valitaan yksi havainto edustamaan klusteria lopulliseen dataan. Valittu havainto voi olla esimerkiksi klusterin keskusta lähinnä oleva havainto. Näin data saadaan tasapainotettua siten, että positiivisia ja negatiivisia havaintoja on yhtä paljon [Lin, W., et al. 2017]. Vastaavasti yliotantamenetelmiin (*engl. oversampling)* kuuluu muun muassa satunnainen yliotanta, jossa puolestaan satunnaisesti kopioidaan vähemmistöluokan havaintoja. Tässä riskinä on mallin ylisovittuminen. [Kotsiantis, et al. 2006].

Tässä tutkielmassa käytetään kahta datatason menetelmää: satunnaista aliotantaa sekä yliotantamenetelmää SMOTEa (*engl. Synthetic Minority Over-sampling TEchnique*) yhdistettynä aliotantaan. Esitellään nämä seuraavaksi.

Aliotanta

Aliotantamenetelmissä vain enemmistöluokan havaintoja karsitaan. Yksinkertaisin menetelmä lienee satunnainen aliotanta, jossa enemmistöluokan havaintoja karsitaan satunnaisesti. Aliotanta voidaan tehdä niin, että otannan jälkeen luokat ovat täysin tasapainossa eli luokkien kokojen välinen suhde on 1:1 tai enemmistöluokan havaintoja voidaan jättää dataan enemmän kuin vähemmistöluokan havaintoja on datassa. Tässä tutkielmassa satunnainen aliotanta kohdistetaan enemmistöluokan muodostaviin ei-tilanneisiin. Testeissä käytetään viittä eri luokkasuhdetta: 1:1, 2:1, 3:1, 4:1 ja 5:1.

SMOTE yhdistettynä aliotantaan

SMOTEssa yliotanta tehdään luomalla synteettisiä vähemmistöluokan havaintoja vähemmistöluokan alkuperäisten havaintojen perusteella. SMOTEssa hyödynnetään *k*:n lähimmän naapurin luokittelijaa. On myös mahdollista lisätä synteettisiä havaintoja SMOTElla ja sen lisäksi vielä vähentää enemmistöluokan havaintoja. Seuraavassa on esitelty pseudokoodilla, miten SMOTE toimii.

Olkoon T vähemmistöluokan havaintojen joukko ja $|T|$ vähemmistöluokan havaintojen lukumäärä. N on prosenttiluku, joka kertoo yliotannan suuruuden, k on lähimpien naapureiden lukumäärä ja m on muuttujien lukumäärä. SMOTE tekee synteettiset havainnot seuraavin askelin (oletus $N \geq 100$ ja N on jaollinen 100:lla):

1. Lasketaan jokaiselle vähemmistöluokan havainnolle (yhteensä $|T|$) k lähintä naapuria.
2. Lasketaan $N' = \left(\frac{N}{100}\right)$
3. Jokaista vähemmistöluokan havaintoa X vasten luodaan N' uutta synteettistä havaintoa seuraavasti:

3.1. Valitaan yksi satunnainen lähin naapuri Y k:n lähimmän naapurin joukosta. Nyt $X = (x_1, x_2, \dots, x_m)$ ja $Y = (y_1, y_2, \dots, y_m)$, missä $x_1, \dots, x_m$ ja $y_1, \dots, y_m$ ovat muuttujien arvoja.

3.2. $d = (y_1, y_2, \dots, y_m) - (x_1, x_2, \dots, x_m) = (y_1 - x_1, y_2 - x_2, \dots, y_m - x_m)$

3.3. Lasketaan satunnaisluku r tasajakaumasta väliltä $[0,1]$

3.4. Uusi synteettinen havainto $S = X + r \cdot d$

3.5. Toistetaan kohdat 3.1 - 3.4 N' kertaa

4. Lopputuloksena algoritmi on luonut $N' \cdot |T|$ uutta synteettistä havaintoa.

Mikäli N on pienempi kuin 100, prosessi on muuten sama, mutta vain N %:sta satunnaisesti valituista vähemmistöluokan havainnoista luodaan synteettinen havainto. [Chawla, et al. 2002]

3.1.2 Algoritmitason menetelmät

Algoritmitason menetelmät ovat nimensä mukaisesti sijoitettu oppimisalgoritmeihin itsessään, kun taas datatason menetelmiä sovelletaan ennen algoritmin soveltamista. Esimerkiksi *k*:n lähimmän luokittelijan kanssa voidaan käyttää painotettuja etäisyyksiä siten, että painoja ei annetakaan tavanomaiseen tapaan yksittäisille havainnoille vaan luokkakohtaisesti niin, että uutta havaintoa luokitellessa naapurustosta löytyy herkemmin myös vähemmistöluokan havaintoja. [Kotsiantis, et al. 2006]

Algoritmitason menetelmissä voidaan lisäksi hyödyntää raja-arvoja. Tietyt algoritmit antavat eksaktin luokan sijaan pistearvon tai todennäköisyyden, joka kuvastaa astetta tai todennäköisyyttä kuulua tiettyyn luokkaan. Näin ollen esimerkiksi luokkajäsenyyden todennäköisyydelle asetettua raja-arvoa vaihtelemalla voidaan tuottaa useita luokittelijoita ja löytää optimaalisin raja-arvo. Tässä tutkielmassa tutkitaan myös raja-arvon vaikutusta muun muassa mallin sensitiivisyyteen ja spesifisyyteen, kun ennusteet jaetaan kahteen luokkaan raja-arvon perusteella: raja-arvoa suuremmat todennäköisyydet muodostavat positiivisen luokan ”tilanneet” ja raja-arvoa pienemmät todennäköisyydet negatiivisen luokan ”ei-tilanneet”.

Kustannus-sensitiivisessä oppimisessa (*engl. cost-sensitive learning*) puolestaan eri luokkien virheellisille ennusteille määritellään eri kustannukset. Tavoitteena on siten minimoida väärän luokittelun kustannus. [Kotsiantis, et al. 2006]

3.1.3 Evaluointimetriikat

Kuten jo todettiin, luokkajakaumaltaan epätasapainoisen datan luokittelijan arvioinnissa tarkkuus (*engl. accuracy*) on harhaanjohtava evaluointimetriikka. Siksi luokkajakaumaltaan epätasapainoisen datan käsittelyyn liittyy oleellisesti oikein valitut evaluointimetriikat. Tarkkuuden sijaan parempia evaluointimetriikoita ovat muun muassa sensitiivisyys, spesifisyys, luotettavuus sekä F_1-pistearvo, jotka esitellään luvussa 3.3.3. Tästä syystä mallien toimivuutta vertaillaan myös näiden mittareiden avulla, vaikka pääpaino evaluointimetriikoissa onkin todennäköisyysennustukseen hyvin soveltuvat logaritminen tappio sekä Brierin pisteet, jotka esitellään luvuissa 3.3.1 ja 3.3.2.

3.2 Luokittelualgoritmit

Sovellusongelman kannalta on mielekkäämpää ennustaa luokkatodennäköisyyksiä luokkien sijaan, sillä varmojen tilaajien sijaan voi olla nimenomaan tärkeää tunnistaa mahdollisia tilaajia. Todennäköisyyksien avulla on myös helppo muuttaa rajaa, jonka perusteella päätetään, kenelle kohdistetaan toimenpiteitä. Monet luokittelualgoritmit sopivat todennäköisyyksien ennustamiseen luonnostaan, kuten päätöspuut, satunnaismetsät ja tukivektorikoneet ja naiivi Bayes-luokittelija. Kuitenkin osa algoritmeista päätyy ennustamaan todennäköisyyksiä kauas ääripäistä 0 ja 1 (kuten tukivektorikoneet) ja osa taas lähellä ääripäitä 0 ja 1 (kuten naiivi Bayes-luokittelija). Satunnaismetsä ennustaa harvoin lähellä ääripäitä olevia ennusteita, koska se yhdistää monen eri mallin ennusteita. Jotta satunnaismetsä ennustaisi todennäköisyyden $p = 0$, täytyisi kaikkien sen päätöspuiden ennustaa todennäköisyys $p = 0$. [Niculescu-Mizil and Caruana. 2005]. Tutkielmassa käytetyt päätöspuualgoritmit C4.5 ja CART pyrkivät luomaan homogeenisiä lehtiä, ja siten todennäköisyysennusteet ovat helpommin lähellä ääripäitä [Elkan and Zadrozny. 2001]. Tähän ongelmaan hyödynnetään kalibrointimenetelmiä, jotka esitellään luvussa 3.4.

Tutkielmassa keskitytään päätöspuualgoritmeihin C4.5 ja CART sekä Breimanin satunnaismetsään. Ensiksi kuitenkin esitellään lyhyesti oleelliset informaatioteorian käsitteet Shannonin entropia (*engl. entropy*), informaatiohyöty (*engl. information gain*), hyötysuhde (*engl. gain ratio*) ja Gini-indeksi, sillä puut – ja siten satunnaismetsä – rakennetaan niiden avulla.

3.2.1 Informaatioteorian käsitteitä

Entropia

Olkoon D harjoitusaineisto, joka sisältää havainnot luokkamuuttujineen. Olkoon luokkamuuttujalla l eri luokkaa, joihin viitataan merkinnöillä C_i, missä $i = 1, \dots, l$. Merkinnät $|C_{i,D}|$ ja $|D|$ viittaavat havaintojen lukumäärään joukoissa. Havainnon luokittelemiseksi tarvittu odotettu informaatio eli aineiston D entropia määritellään seuraavasti:

$$entropia(D) = -\sum_{i=1}^{l} p_i log_2(p_i),$$

missä p_i on todennäköisyys sille, että satunnainen havainto aineistossa D kuuluu luokkaan C_i. Todennäköisyys määritetään laskemalla $\frac{|C_{i,D}|}{|D|}$. [Han and Kamber. 2001]

Informaatiohyöty

Kun määritetään informaatiohyötyä muuttujalle A, tarvitaan arvo, joka kertoo, kuinka paljon informaatiota vielä tarvitaan havainnon luokittelemiseksi, kun aineisto on ositettu muuttujaa A käyttäen. Aineiston ositus määritetään A:n arvojen perusteella. Mikäli muuttuja on diskreetti, eri ositukset voidaan tehdä suoraan muuttujan arvojen perusteella. Jos muuttujan A arvoja on v kappaletta $\{a_1, a_2, \dots, a_v\}$, aineisto D ositetaan muuttujan arvojen perusteella osajoukkoihin $\{D_1, D_2, \dots, D_v\}$, missä $D_i \cap D_j = \emptyset \; \forall \; i \neq j$ ja $i, j \leq v$. Havainnon luokitteluun tarvittavan informaation määrä osituksen jälkeen on

$$entropia_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times entropia(D_j).$$

Mitä pienempi tämä entropia on, sitä enemmän ositukset sisältävät vain yhteen luokkaan kuuluvia havaintoja. Tähän viitataan termillä puhtaus (*engl. purity*). Varsinainen informaatiohyöty saadaan laskettua kahden edellisen kaavan perusteella

$$informaatiohyöty(A) = entropia(D) - entropia_A(D).$$

Informaatiohyödyn avulla voidaan mitata muuttujien tärkeyttä aineistossa ja järjestää muuttujat tärkeyden mukaan. Informaatiohyödyltään suurin muuttuja on luokkamuuttujan suhteen kaikista erottelevin muuttuja aineistossa. [Han and Kamber. 2001]

Hyötysuhde

Informaatiohyödyn avulla voidaan laskea hyötysuhde, joka ottaa huomioon lisäksi muuttujien eri arvojen lukumäärän. Pelkän informaatiohyödyn käyttäminen suosii helposti muuttujia, jotka saavat paljon eri arvoja. Karkein esimerkki on jonkinlainen yksilöllinen tunnistemuuttuja, joka saa vain uniikkeja arvoja. Tällaiselle muuttujalle ehdollinen entropia $entropia_A(D) = 0$, johtuen siitä, että muuttujien arvojen perusteella tehdyt ositukset sisältävät vain yhden havainnon, joka luonnollisesti kuuluu vain yhteen luokkaan. Kuitenkin tämä on luokittelun kannalta täysin turha muuttuja. Hyötysuhde saadaan jakamalla informaatiohyöty niin kutsutulla ositusinformaatiolla [Han and Kamber. 2001]:

$$ositusinformaatio = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times log_2\left(\frac{|D_j|}{|D|}\right).$$

Eli

$$hyötysuhde = \frac{informaatiohyöty(A)}{ositusinformaatio(A)}.$$

Gini-indeksi

Gini-indeksi mittaa sitä, kuinka usein sattumanvaraisesti valittu havainto luokitellaan väärin, mikäli se luokitellaan luokkatodennäköisyyksien mukaan. Kullekin muuttujalle lasketaan Gini-indeksi määrittelemällä ensin muuttujan arvoihin perustuvien aineiston ositusten Gini-indeksit kaavalla

$$Gini = 1 - \sum_{i=1}^{l} (p_i)^2,$$

missä l on luokkien lukumäärä ja p_i on niiden havaintojen osuus, joiden luokka on i.

Kun kullekin muuttujan arvolle on laskettu Gini-indeksit, lasketaan koko muuttujalle Gini-indeksi näiden arvokohtaisten Gini-indeksien painotettuna summana, jossa paino määritellään [Hssina, et al. 2014]

$$paino = \frac{osajoukon\ koko}{aineiston\ koko}.$$

3.2.2 Päätöspuu C4.5

Quinlan [1993] esittelee kirjassaan laaja-alaisesti kehittämäänsä C4.5 algoritmia, joka on parannettu versio Quinlanin aiemmin kehittämästä ID3-algoritmista. C4.5 mahdollistaa jatkuvien muuttujien käytön ja se pystyy käsittelemään puuttuvia arvoja, kun taas ID3 oletti muuttujien olevan nominaaliasteikoilla ilman puuttuvia arvoja. Mikäli muuttuja sisältää joitakin puuttuvia arvoja, hyötysuhteen laskennassa ei huomioida näitä havaintoja C4.5 päätöspuuta rakennettaessa. [Hssina, et al. 2014].

Jatkuville muuttujille etsitään optimaaliset osituskohdat. Muuttujan arvot järjestetään suuruusjärjestykseen ja vuorollaan testataan jokaista arvoa osituskohtana. Se kohta, joka maksimoi hyötysuhteen, valitaan lopulliseksi osituskohdaksi. [Hssina, et al. 2014]

ID3 on erittäin herkkä suosimaan muuttujia, jotka saavat useita eri arvoja. C4.5 tuo tähän parannuksen ottamalla huomioon myös sen, montako eri arvoa muuttuja sisältää ja käyttää muuttujien valinnassa informaatiohyödyn sijaan hyötysuhdetta.

Neljäntenä parannuksena esitetään puun karsiminen mallin luonnin jälkeen. Päätöspuiden yleisenä ongelmana on harjoitusaineiston ylisovitus, jolloin uusien havaintojen luokittelu vaikeutuu. Päätöspuun karsimisella pyritään sekä vähentämään luodun mallin monimutkaisuutta sekä parantamaan ennustuksen tarkkuutta. Näihin päästään, kun ylisovittamista pienennetään sellaisten osien poistamisella, jotka ovat todennäköisesti rakentuneet attribuuttien arvoissa olevien virheiden perusteella. Tästä huolimatta C4.5 on melko herkkä poikkeaville arvoille. [Hssina, et al. 2014]

Esitellään seuraavaksi vaiheittain [S. Ruggieri. 2002], miten päätöspuu muodostetaan C4.5-algoritmilla. Olkoon T havaintojen joukko, joka on yhteydessä muodostettavaan solmuun. Koska puuta rakennetaan juuresta käsin, aluksi T on koko harjoitusaineisto, jolla puu muodostetaan. Karsiminen (kohta 8) puolestaan aloitetaan lehdistä edeten kohti juurta.

1. Lasketaan jokaisen luokan frekvenssi joukossa T.

2. Jos kaikki havainnot joukossa T kuuluvat samaan luokkaan C_i tai havaintoja on alle määritellyn rajan, muodostetaan lehtisolmu N. Lehden luokitteluvirhe on niiden havaintojen painotettu summa, jotka eivät kuulu luokkaan C_i. Muutoin jatketaan puun muodostamista tästä kohdasta.

3. Lasketaan jokaisen muuttujan informaatiohyötysuhde. Diskreeteille muuttujille hyötysuhde lasketaan muuttujien arvojen mukaisella osituksella. Jatkuvien muuttujien kohdalla etsitään optimaalisin jakokohta käyttämällä jokaista *A*:n arvoa jakokohtana *R*, jolloin muodostuvat joukot havainnoista, joissa $\leq R$ ja $> R$ ja valitsemalla ositus, joka maksimoi hyötysuhteen. (ks. luku 3.2.1)

4. Valitaan hyötysuhteeltaan suurin muuttuja *A* solmuun *N*.

5. Jos valittu muuttuja *A* on jatkuva, lasketaan raja-arvo, jonka kohdalta havainnot jaotellaan. Raja-arvo on koko harjoitusaineiston muuttujan suurin arvo, joka on pienempi kuin kohdassa 3 saatu paikallinen raja-arvo.

6. Haarautetaan solmu *N*. Jatkuvan muuttujan tapauksessa uusia lapsisolmuja muodostuu kaksi lasketun raja-arvon perusteella ja diskreetin muuttujan tapauksessa sen eri arvojen lukumäärän verran. Jokaiseen haaraan *j* liittyy havaintojen joukko T_j.

7. Käydään jokainen joukko T_j läpi. Mikäli T_j on tyhjä, muodostetaan lehti, jonka luokaksi tulee lehden vanhemmassa useimmiten esiintynyt luokka. Muutoin toistetaan joukon T_j kohdalla rekursiivisesti kohdat 1–7. Joukkoon T_j lisätään lisäksi kaikki ne havainnot, joilla muuttujan *A* arvo on tuntematon.

8. Puun karsiminen: Lasketaan solmun luokitteluvirhe. Solmun luokitteluvirhe on sen lapsisolmujen luokitteluvirheiden summa. Lasketaan lisäksi virhe, joka saadaan luokittelemalla kaikki havainnot joukossa *T* yleisimpään luokkaan. Jos solmun luokitteluvirhe on suurempi kuin tämä virhe, muutetaan solmu lehdeksi ja poistetaan sen kaikki alipuut.

3.2.3 Päätöspuu CART

Breimanin ja kumppaneiden [1984] kehittämän päätöspuumenetelmän nimi on CART (*engl. Classification and Regression Trees*). Algoritmilla voi nimensä mukaisesti

rakentaa sekä luokittelu- että regressiomalleja. CART-binääripäätöspuu rakennetaan samalla tavoin juuresta käsin kuten C4.5, mutta muuttujien valinnassa käytetään monesti Gini-indeksiä, joka esiteltiin luvussa 3.2.1. Toinen vaihtoehto on käyttää kahtiajakokriteeriä (*engl. twoing criteria*). Gini-indeksi saattaa tuottaa melko pieniä, mutta puhtaita solmuja, kun taas kahtiajakokriteeri huomioi puhtauden lisäksi myös solmun koon ja pyrkii luomaan suunnilleen samankokoisia solmuja [Breiman. 1996]. CART käsittelee numeerista ja kategorista dataa sekä käsittelee puuttuvia arvoja, eikä se ole erityisen herkkä poikkeaville arvoille. Karsintavaiheessa käytetään kustannus–kompleksisuus-karsintametodia, jota ei tässä esitellä tarkemmin.

Esitellään seuraavaksi askeleet, joilla CART-päätöspuu muodostetaan [Yohannes and Webb. 1999, Steinberg and Colla. 1995, Lewis. 2000]. Olkoon T havaintojen joukko, joka on yhteydessä muodostettavaan solmuun. Koska puuta rakennetaan juuresta käsin, T on aluksi koko harjoitusaineisto.

1. Etsitään jokaisen muuttujan paras jakokohta. Jatkuvien muuttujien kohdalla etsitään optimaalisin jakokohta käyttämällä jokaista muuttujan arvoa jakokohtana R, jolloin muodostuvat joukot havainnoista, joissa $\leq R$ ja $> R$ ja valitsemalla se, joka minimoi Gini-indeksin. Diskreettien muuttujien kohdalla tehdään kaikki jakotavat, millä T voidaan jakaa kahteen joukkoon ja valitaan se jakotapa, joka minimoi Gini-indeksin, jolloin epäpuhtaus (*engl. impurity*) vähentyy eniten.

2. Edellisen kohdan perusteella valitaan solmuun N muuttuja, joka minimoi Gini-indeksin. Määritetään solmun luokaksi solmuun liittyvien havaintojen enemmistöluokka. Oletuksena kaikilla luokilla on yhtä suuret väärin luokittelun kustannukset.

3. Haarautetaan solmu N kahteen haaraan valitun muuttujan perusteella.

4. Jos lopettamiskriteeri ei täyty, toistetaan solmun N kahdella lapsella kohdat 1—3. CART jatkaa puun rakentamista niin pitkälle, kunnes jokainen havainto muodostaa lehden tai kun lehdessä on enää vähän havaintoja, esimerkiksi 10.

5. Karsitaan puu kustannus–kompleksisuus-metodilla ja valitaan lopuksi optimaalisin puu.

Mikäli puuhun valitulla muuttujalla on puuttuvia arvoja, puuttuvan arvon sisältäviä havaintoja ei jätetä sivuun, vaan niitä käsitellään parhaan sijaismuuttujan avulla. Sijaismuuttuja jäljittelee tai ennustaa varsinaisen puuhun valitun ensisijaisen muuttujan jakokohtaa. Sijaismuuttujan jakokohta voi erota ensisijaisen muuttujan jakokohdasta, mutta havaintojen lukumäärä vasemmassa ja oikeassa lapsisolmussa on oltava hyvin lähellä samaa, mitä ensisijaisen muuttujan jakokohta tuottaa. Kun ensisijaisen muuttujan arvo puuttuu, käytetään tätä sijaismuuttujaa sen päättelemiseen, kuuluuko havainto vasemmalle vai oikealle puolelle. [Yohannes and Webb. 1999]

3.2.4 Satunnaismetsä

Satunnaismetsä koostuu useasta eri päätöspuusta, jotka on muodostettu toisistaan riippumatta. Luokittelussa havainnon luokaksi määritellään yleisin luokka, jonka yksilölliset puut antavat ja regressiomallin tapauksessa ennusteeksi tulee yksittäisten puiden antamien ennusteiden keskiarvo. Breiman [2001] yhdisti oman ideansa havaintojen satunnaisotannasta (*engl. bagging)* Hon [1998] jo aiemmin esittelemään ideaan muuttujien valinnasta. Tyypillisesti yhden puun harjoitusdataan valitaan $\sqrt{m}$ muuttujaa, missä m on muuttujien lukumäärä.

Jokainen satunnaismetsän puu rakennetaan siis erilaisella harjoitusdatalla. Harjoitusdata muodostetaan valitsemalla satunnaisesti havaintoja (otanta palauttaen) ja satunnaisesti muuttujia (otanta palauttamatta) ennalta sovitut määrät. Harjoitusdatan ulkopuolelle jää siis joukko havaintoja, joita ei käytetä puun koulutukseen, mutta joukolla on tehtävä mallin evaluoinnissa. Tähän joukkoon viitataan yleisesti termillä OOB-havainnot (OOB tulee englannin kielen sanoista *out-of-bag).* Havaintojen otanta palauttaen vähentää mallin varianssia. Vaikka yksittäinen puu on herkkä poikkeaville arvoilla ja helposti ylisovittaa harjoitusdataa, eri harjoitusdatoilla koulutettujen päätöspuiden yhdistelmä ei tee tätä, kunhan puut eivät korreloi voimakkaasti. Muuttujien otanta puolestaan vaikuttaa puiden väliseen korrelaatioon. Mikäli muuttujien otantaa ei tehtäisi ja datassa olisi muutama luokkamuuttujaa vahvasti ennustava muuttuja, tulisivat ne valituksi useaan satunnaismetsän päätöspuuhun. Tällöin puut korreloisivat keskenään vahvasti. [Breiman. 2001]

Alla esitellään lyhyesti satunnaismetsän muodostaminen. Olkoon muuttujien joukko $A_1, \dots, A_m$, havaintojen joukko $X_1, \dots, X_n$ ja satunnaismetsän puiden lukumäärä N. Kohta 3 vastaa luvussa 3.2.3 esitettyä päätöspuun CART rakentamistapaa.

1. Otetaan havainnoista osajoukko *X'* suorittamalla otanta palauttaen.
2. Otetaan satunnaisesti muuttujista muuttujien joukko *A'* suorittamalla otanta palauttamatta.
3. Muodostetaan päätöspuu muuttujilla *A'* ja havainnoilla *X'*.
4. Toistetaan vaiheita 1–3, kunnes *N* päätöspuuta on muodostettu. Satunnaismetsä on näin muodostettujen puiden kokoelma.

Ennustettaessa ennennäkemättömän havainnon luokkaa, määritetään jokaisen yksittäisen puun tuottama luokka havainnolle. Satunnaismetsän tuottamaksi ennusteeksi valitaan se, mitä suurin osa yksittäisistä päätöspuista ennusti. [Breiman. 2001]

Breimanin satunnaismetsä tuottaa metriikkoja, jotka kertovat muuttujien tärkeydestä mallissa: keskivähenemä tarkkuudessa (*engl. Mean Decrease Accuracy, MDA)* ja keskivähenemä Gini-indeksissä (*engl. Mean Decrease Gini, MDG)*. Koska MDG:n on osoitettu suosivan tietynlaisia muuttujia [Strobl, et al. 2007, Boulesteix, et al. 2011], on sitä syytä käyttää varoen. Tutkielmassa ei hyödynnetä MDG:ta vaan keskitytään vain MDA-metriikkaan.

MDA

MDA perustuu virhemetriikkaan, joka on luokittelutehtävän yhteydessä virhesuhde (*engl. error rate)* ja regression yhteydessä keskineliövirhe (*engl. mean squared error*). MDA määritellään seuraavasti:

$$MDA_j^M = \frac{1}{N}\sum_{t=1}^{N}\left(MP_{tj} - M_{tj}\right),$$

missä

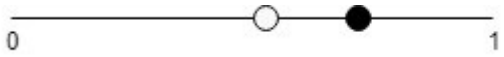
- N on päätöspuiden lukumäärä satunnaismetsässä
- M_{tj} on puun t OOB-havaintojen avulla laskettu virhe ennen ennustavan muuttujan X_j arvojen permutaatiota
- MP_{tj} on puun t OOB-havaintojen avulla laskettu virhe sen jälkeen, kun ennustavan muuttujan X_j arvot ovat satunnaisesti permutoitu.

Ennustavan muuttujan X_j tärkeyttä arvioidaan siis seuraavasti: mikäli X_j ei ole yhteydessä ennustettavaan muuttujaan, sen arvojen permutaatio ei vaikuta luokitteluun eikä sillä siten ole juurikaan vaikutusta puun virheeseen. Tällöin M_{tj} ja MP_{tj} ovat lähellä toisiaan ja muuttujan tärkeys on siten lähellä nollaa. Jos taas X_j on yhteydessä ennustettavaan muuttujaan, sen arvojen permutaatio vaikuttaa luokitteluun ja sen poisjättäminen mallista suurentaa virhettä, jolloin $MP_{tj} > M_{tj}$. Tällöin MDA on positiivinen. Mitä suurempi MDA:n arvo, sitä tärkeämpi muuttuja on mallissa. [Janitza, et al. 2016]

3.3 Evaluointi

Koska tutkielmassa ennustetaan todennäköisyyksiä, on tulosten evaluointimetriikoiksi valittu logaritminen tappio (*engl. logarithmic loss, logistic loss tai logloss*) ja Brierin pisteet (*engl. Brier score*). Nämä metriikat ottavat huomioon ennustuksen virheen suuruuden. Tämän lisäksi raportoidaan sekaannusmatriisi ja siitä saatavat evaluointimetriikat: tarkkuus, sensitiivisyys, spesifisyys, luotettavuus ja F_1-pistearvo. Sekaannusmatriisi muodostetaan muutamalla potentiaalisella raja-arvolla, jotta voidaan tutkita raja-arvon vaikutusta ennustetarkkuuteen.

Sekä logaritmiselle tappiolle että Brierin pisteille pätee, että mitä pienempi arvo sen parempi malli. Kuvassa 13 musta ja valkoinen pallo merkitsevät kahta eri todennäköisyysennustetta. Mikäli todellinen luokka on 1, musta ennuste on lähempänä todellisuutta ja saa siten pienemmän arvon logaritmisella tappiolla ja Brierin pisteillä mitattuna.



Kuva 13: Ennusteiden paremmuunjärjestys

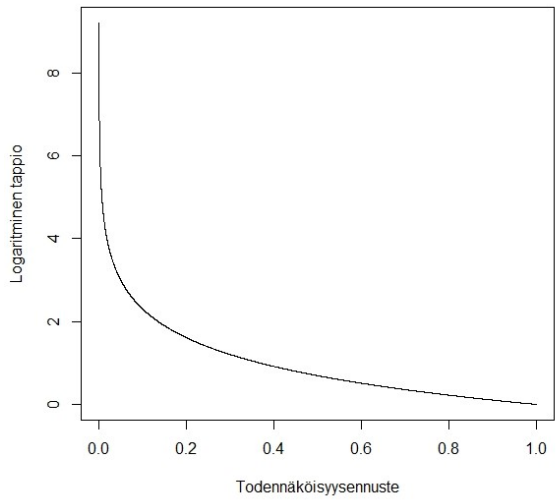
3.3.1 Logaritminen tappio

Logaritminen tappio saadaan laskemalla ensin jokaiselle havainnolle erikseen logaritminen tappio ja ottamalla tästä keskiarvo. Yhdelle havainnolle saadaan binääriluokittelussa logaritminen tappio seuraavasti, kun $p \neq 0$:

$$logaritminen\ tappio = -(y\,log_2(p) + (1-y)\,log_2(1-p)),$$

missä y tarkoittaa luokkatunnusta ja p ennustettua todennäköisyyttä. Oletetaan, että positiiviseen luokkaan viitataan luokkatunnuksella 1 ja negatiiviseen luokkaan luokkatunnuksella 0. Jos $p = 0$, voidaan todennäköisyydeksi asettaa hyvin pieni arvo, esimerkiksi $p = 10^{-15}$, jolloin logaritminen tappio voidaan laskea.

Täydellisellä mallilla logaritminen tappio on siis 0. Jos oikea luokkatunnus on 1, logaritminen tappio kasvaa voimakkaasti, kun lähestytään ennusteissa nollaa. Logaritmisen tappion arvot eri ennustetuilla todennäköisyyksillä näkyvät kuvassa 14. [Deep Learning Course Wiki. 2017]



Kuva 14: Logaritminen tappio eri ennustearvoilla, kun todellinen luokka on positiivinen

3.3.2 Brierin pisteet

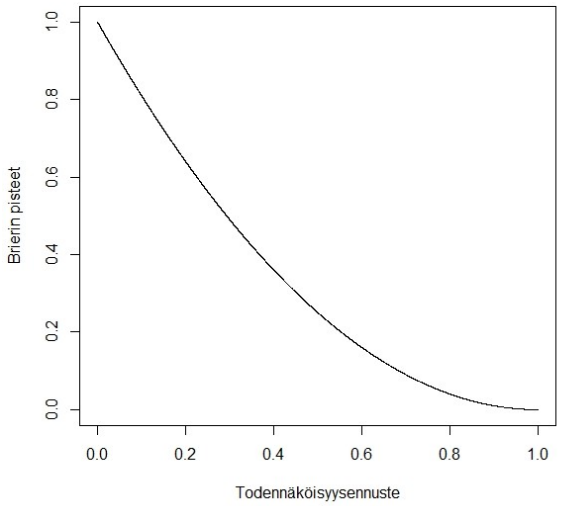
Glenn W. Brier [1950] esitteli mukaansa nimetyn Brierin pisteet sääennusteiden arvioimiseen. Brierin pisteet lasketaan seuraavasti:

$$BS = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2,$$

missä n on havaintojen lukumäärä, p_i on ennustettu todennäköisyys ja y_i on oikea luokka havainnolle X_i. Itse asiassa kyseessä ennusteen keskimääräinen neliövirhe.

Vastaavasti logaritmisen tappion kanssa, täydellisellä mallilla Brierin pisteet ovat 0. Mikäli taas kaikkien havaintojen ennusteiksi määritetään 0,5, saadaan Brierin pisteiksi 0,25, sillä riippumatta todellisesta luokasta yksittäisen havainnon Brierin pisteet ovat

$(0{,}5-1)^2 = (0{,}5-0)^2 = 0{,}25$. Brierin pisteiden arvot yksittäisellä positiivisella havainnolla eri ennustetuilla todennäköisyyksillä näkyvät kuvassa 15.



Kuva 15: Brierin pisteet eri ennustearvoilla, kun todellinen luokka on positiivinen

3.3.3 Sekaannusmatriisi ja evaluointimetriikat

Tässä luvussa esitellään sekaannusmatriisi ja siitä saatavat metriikat tarkkuus, sensitiivisyys ja spesifisyys. Tarkkuus on tyypillisin luokittelijan hyvyyttä kuvaava metriikka, joka kertoo, moniko havainto on luokiteltu oikein. Sensitiivisyys on todennäköisyys sille, että positiivinen havainto luokitellaan positiiviseksi ja spesifisyys vastaavasti todennäköisyys sille, että negatiivinen havainto luokitellaan negatiiviseksi.

		Todelliset	
		positiivinen	negatiivinen
Ennusteet	pos.	Oikea positiivinen, TP	Väärä positiivinen, FP
	neg.	Väärä negatiivinen, FN	Oikea negatiivinen, TN

Kuva 16: Sekaannusmatriisin idea

Sekaannusmatriisissa (kuva 16) on kyse ristiintaulukoinnista ja sen solut ovat frekvenssejä. Esimerkiksi solu ”oikea positiivinen” sisältää niiden havaintojen lukumäärän, joiden ennustettu ja todellinen luokka ovat positiivisia. Sekaannusmatriisista

saadaan helposti tarkkuuden (*engl. accuracy*), sensitiivisyyden ja spesifisyyden määritelmät. Tarkkuus tarkoittaa oikein ennustettujen osuutta:

$$tarkkuus = \frac{TN + TP}{TN + FP + FN + TP}.$$

Sensitiivisyys on todennäköisyys, että positiivinen tapaus tunnistetaan positiiviseksi:

$$sensitiivisyys = \frac{TP}{TP + FN}.$$

Vastaavasti spesifisyys on todennäköisyys, että negatiivinen tapaus tunnistetaan negatiiviseksi:

$$spesifisyys = \frac{TN}{TN + FP}.$$

Tarkkuus on tyypillinen mallin toimivuutta kuvastava metriikka. Kuitenkin luokkajakaumaltaan epätasapainoisen datan kanssa tarkkuus on ongelmallinen. Mikäli enemmistöluokka käsittää 99 % kaikista havainnoista, pelkästään arvaamalla jokaisen testiaineiston havainnon enemmistöluokan havainnoksi, saadaan tarkkuudeksi huikea 0,99. Kuitenkin tällainen luokittelija on hyödytön. On tärkeämpää keskittyä positiivisten havaintojen havaitsemiseen, joten esitellään sensitiivisyyden lisäksi metriikka luotettavuus (*engl. precision)*. Luotettavuus kuvastaa, kuinka usein positiiviseksi ennustettu tapaus on myös todellisuudessa positiivinen eli täsmällisesti ilmaistuna:

$$luotettavuus = \frac{TP}{TP + FP}.$$

Jos halutaan maksimoida yhtäaikaisesti luotettavuus ja sensitiivisyys, voidaan hyödyntää F_1-pistearvoa (*engl. F_1-score, F-score*), joka määritellään seuraavasti:

$$F_1 = \frac{sensitiivisyys \times luotettavuus}{sensitiivisyys + luotettavuus}.$$

Kyseessä on luotettavuuden ja sensitiivisyyden harmoninen keskiarvo. [He and Garcia. 2009]

3.4 Kalibrointimenetelmät

Otantamenetelmät ovat yksi suosituimmista keinoista luokkajakaumaltaan epätasapainoisen datan hallintaan. Erityisesti aliotanta on yksinkertaisuudessaan käyttökelpoinen metodi: enemmistöluokan havaintoja poistetaan satunnaisesti haluttu määrä siten, että datassa on luokittelualgoritmille sopivammassa suhteessa negatiivisia ja positiivisia havaintoja. Tämä ei kuitenkaan ole ongelmatonta, sillä alkuperäisen harjoitusaineiston muokkaaminen vaikuttaa helposti mallin tarkkuuteen, kun mallia testataan alkuperäisellä jakaumalla olevalla testidatalla.

Kun harjoitusaineiston havaintoja karsitaan, luokittelijan varianssi kasvaa. Luokittelija, jolla on suuri varianssi, on liian tarkka malli ja siten ylisovittaa harjoitusdataa. Samalla, kun havaintoja poistetaan, muutetaan myös luokkasuhdetta. Tällöin prioritodennäköisyydet muuttuvat, ja se vaikuttaa myös ennustettujen todennäköisyyksien jakaumaan vääristäen niitä. Ensimmäistä ongelmaa voidaan korjata *bagging*-tekniikoilla ja jälkimmäistä ongelmaa kalibroinnilla. [Dal Pozzolo, et al. 2015]. Seuraavaksi esitellään kaksi tutkielmassa käytettävää kalibrointitekniikkaa: Plattin skaalaus ja isotoninen regressio. Kalibroinnin ideana on muuntaa tietyin säännöin ennustettuja ja otannan vuoksi vääristyneitä todennäköisyyksiä mallin tarkkuuden parantamiseksi.

3.4.1 Plattin skaalaus

Platt [1999] esitteli nimeään kantavan kalibrointimenetelmän alun perin tukivektorikoneille. Plattin skaalaus perustuu logistiseen regressioon. Olkoon f koulutettu malli ja siten havainnon X numeerinen ennuste $f(X)$. Plattin skaalauksella saadaan havainnon kalibroitu ennuste seuraavasti:

$$P(y = 1|X) = \frac{1}{1 + e^{(Af(X)+B)}},$$

missä A ja B ovat algoritmin avulla optimoidut vakiot.

Olkoon harjoitusdatan koko n ja havainnot X_i, missä $i = 1, \dots, n$. Vakioiden A ja B optimoimiseen käytetty harjoitusdata sisältää mallilla f saadut ennusteet $f(X_i)$ ja todelliset luokat y_i eli parit $(f(X_i), y_i)$. Todelliset luokat y_i saivat arvon -1 tai 1, ja ne muutetaan arvoiksi 0 tai 1:

$$t_i = \frac{y_i + 1}{2},$$

jolloin saamme lopullisen harjoitusdatan, joka sisältää parit $(f(X_i),\ t_i)$.

Vakiot A ja B saadaan minimoimalla harjoitusdatan negatiivinen logaritminen uskottavuus:

$$-\sum_{i=1}^{n}(t_i \log(p_i) + (1 - t_i)\log(1 - p_i)),$$

missä

$$p_i = \frac{1}{1 + e^{(Af(X_i)+B)}}.$$

Tähän kaksiparametriseen minimointitehtävään Platt käytti algoritmia, joka perustuu Levenberg–Marquardt-algoritmiin [Press, et al. 1992]. Myöhemmin on kuitenkin esitetty parempi algoritmi parametrien A ja B löytämiseksi, Newtonin metodi [Lin, H., et al. 2007].

Näiden vakioiden optimointiin on syytä käyttää harjoitusdatasta erillistä dataa ylisovituksen ja järjestelmällisen harhan välttämiseksi. Yksi vaihtoehto on käyttää harjoitusdatasta erotettua validointidataa, jota ei käytetä itse mallin koulutukseen vaan ainoastaan parametrien A ja B löytämiseen.

Erotettua validointidataa parempi vaihtoehto on käyttää ristiinvalidointia (luku 3.5), jossa harjoitusdata jaetaan kutakuinkin yhtä suuriin osiin. Esimerkiksi 3-kertaisessa ristiinvalidoinnissa osiin T_1, T_2 ja T_3. Jokaisella kolmesta kierroksesta kahdella osalla koulutetaan malli ja kolmannella jäljellä jäävällä osalla saadaan ennusteet $f_i(T_i)$. Täten saadaan kolme eri mallia f_1, f_2 ja f_3 ja niiden tuottamat ennusteet $f_1(T_1)$, $f_2(T_2)$ ja $f_3(T_3)$. Yhdistämällä nämä ennusteet ja todelliset luokat muutettuna, saadaan harjoitusdata parametrien A ja B löytämiseen.

Ristiinvalidoinnista huolimatta ylisovitus on mahdollista. Tämän takia Platt suositteli, että arvojen $t = 0$ ja $t = 1$ sijaan käytettäisiin positiiviselle ja negatiiviselle luokalle muunnoksia:

$$t_+ = \frac{n_+ + 1}{n_+ + 2}$$

ja

$$t_- = \frac{1}{n_- + 2},$$

missä n_+ on positiivisten havaintojen lukumäärä ja n_- vastaavasti negatiivisten havaintojen lukumäärä. [Platt. 1999]

3.4.2 Isotoninen regressio

Isotoninen regressio [Robertson, et al. 1988] on ei-parametrinen muoto regressiosta. Isotoninen regressio täyttää kaksi ehtoa: käyrä on isotoninen eli ei-laskeva kaikkialla ja lisäksi se sovittuu mahdollisimman lähelle havaintopisteitä. Tyypillinen algoritmi isotonisen regression laskemiseen on PAV (lyhenne englannin kielisestä termistä *pair-adjacent violators),* joka käyttää keskineliövirhe-kriteeriä [Elkan and Zadrozny. 2002].

PAV etsii isotonisen ratkaisun seuraavasti. Olkoon $\{X_i\}$ havaintojen joukko, missä $i = 1,..,n$ ja n on joukon koko, $g(X_i)$ on ennustettavan muuttujan *i*. arvo ja g' on isotoninen regressiofunktio, jota etsitään. Jos g on jo isotoninen, niin asetetaan isotoniseksi regressiofunktioksi $g'(X_i) = g(X_i)$. Mikäli g ei ole isotoninen, on olemassa vähintään yksi perättäinen havaintopari, joka rikkoo isotonisuuden eli $g(X_{i-1}) > g(X_i)$. Tällöin $g(X_{i-1})$ ja $g(X_i)$ korvataan näiden keskiarvolla ja isotonisuusehto pätee näiden havaintojen kohdalla. Jos uusi, kooltaan $n-1$, arvojen joukko on tämän jälkeen isotoninen, saadaan isotoniseksi regressiofunktioksi $g'(X_{i-1}) = g'(X_i) = \big(g(X_{i-1}) + g(X_i)\big)/2$ ja muutoin $g'(X_j) = g(X_j)$. Tätä menettelytapaa jatketaan, kunnes on saatu isotoninen arvojen joukko. PAV palauttaa siis lopullisena tuloksena joukon välejä ja jokaiselle välille *k* estimaatin $g'(k)$ siten, että $g'(k+1) \geq g'(k)$. [Elkan and Zadrozny. 2002]

Tutkielmassa isotoninen regressio, kuten Plattin skaalauskin, tuotetaan erillisen validaatiodatan avulla, jota ei ole käytetty mallin varsinaisessa koulutuksessa. Validaatiodata ajetaan ensin koulutetun mallin läpi ennusteiden saamiseksi. Tämä on joukko $\{X_i\}$. Lisäksi tunnetaan havaintojen todelliset luokat $g(X_i)$ eli negatiivisilla havainnoilla arvo on 0 ja positiivisilla arvo on 1. Havainnot järjestetään ennusteiden mukaan kasvavaan järjestykseen. Tämän jälkeen PAV laskee isotonisen regression edellisessä kappaleessa esitetyllä tavalla. Mikäli saadut ennusteet järjestävät havainnot täydellisesti eli kaikki negatiiviset havainnot ovat ennen positiivisia, g on jo isotoninen ja negatiivisten havaintojen uudeksi todennäköisyydeksi tulee 0 ja vastaavasti positiivisten havaintojen luokaksi tulee 1. Kun testidatan havainnosta luodaan ennuste, luodaan siitä ensin normaalisti mallilla ennuste, minkä jälkeen hyödynnetään isotonista regressiofunktiota etsimällä ensin väli *k*, johon saatu ennuste kuuluu ja palauttamalla välille löydetyn estimaatin $g'(k)$. Arvo $g'(k)$ on siten isotonisella regressiolla saatu kalibroitu ennuste.

3.4.3 Kalibrointikuvio

Kalibrointikuvion avulla voidaan arvioida, kuinka luotettavia saadut todennäköisyysennusteet ovat. Kalibrointikuvio piirretään kalibroinnissa käytetyn datan

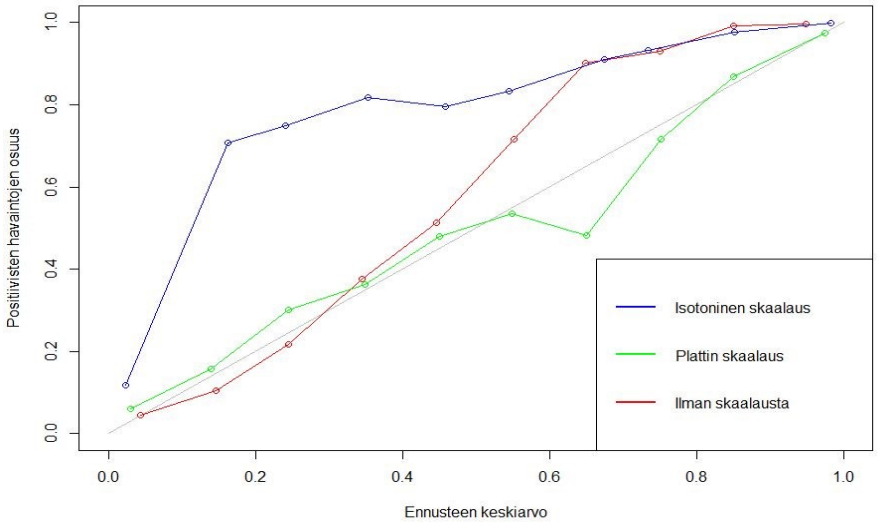
avulla. Jokaista datan havaintoa X_i vastaa ennuste e_i ja todellinen luokka y_i. Ennusteen e_i voidaan ajatella olevan luotettava, jos sitä vastaavien todellisten luokkien positiivisten havaintojen osuus on yhtä suuri kuin ennuste e_i. [Bröcker and Smith. 2007]. Olkoon esimerkiksi neljä havaintoa, joilla kaikilla on ennuste $e_i = 0{,}25$, kun havaintojen luokat ovat 0, 0, 0 ja 1. Tällöin positiivisten havaintojen osuus on $0{,}25$, joten ennuste e_i on luotettava. Mikäli vastaavat luokat olisivat 1, 1, 1 ja 1, positiivisten havaintojen osuudeksi tulisi 1, eikä ennusteta e_i voida siten pitää luotettavana.

Numeerisia ennustuksia tehdessä yhtä ennustetta ei välttämättä vastaa montaa havaintoa. Tämän vuoksi ennusteet jaotellaan arvovälien määrittelemiin ryhmiin. Esimerkiksi tasavälein jaoteltuna ryhmät voivat olla $B_1 = \{e_i \in [0, 0{,}1]\}, \dots, B_{10} = \{e_i \in [0{,}9, 1]\}$. Tällöin ennusteet e_i ryhmässä B ovat luotettavia, mikäli niitä vastaavien todellisten luokkien y_i positiivisten havaintojen osuus on sama kuin ennusteiden $e_i \in B$ keskiarvo. [Bröcker and Smith. 2007]
Varsinainen kalibrointikuvio piirretään seuraavalla tavalla. Ensin ennusteet e_i jaotellaan ryhmiin B_k, $k = 1, \dots, K$. Sen jälkeen havainnot X_i määritetään ryhmiin B_k niiden saamien ennusteiden perusteella. Havainnot X_i ryhmässä B_k muodostavat joukon I_k. Joukon I_k positiivisten havaintojen osuus pos_k saadaan laskemalla kaikki positiiviset tapaukset joukossa I_k ja jakamalla ne koko joukon I_k koolla:

$$pos_k = \frac{\sum_{i \in I_k} y_i}{|I_k|}.$$

Ryhmää B_k vastaamaan määritetään yksittäinen piste x-akselille. Pisteen on tarkoitus esittää ryhmän B_k tyypillistä arvoa. Se voi olla esimerkiksi luokan keskikohta tai ennusteiden keskiarvo ryhmässä B_k. Tässä tutkielmassa kalibrointikuviot piirretään keskiarvoa hyödyntäen. [Bröcker and Smith. 2007]

Kuvassa 17 on tutkielmassa koulutetun mallin kalibrointikuvio. Kalibroimattomat ennusteet on esitetty punaisella värillä, Plattin skaalauksella kalibroidut ennusteet vihreällä värillä ja isotonisella regressiolla kalibroidut ennusteet sinisellä värillä. Kuvan perusteella Plattin skaalaus näyttää tuottavan luotettavimmat todennäköisyysennusteet, koska se mukailee parhaiten täydellistä kalibraatiota noudattavaa harmaata suoraa.



Kuva 17: Esimerkki kalibrointikuviosta

3.5 *K*-kertainen ristiinvalidointi

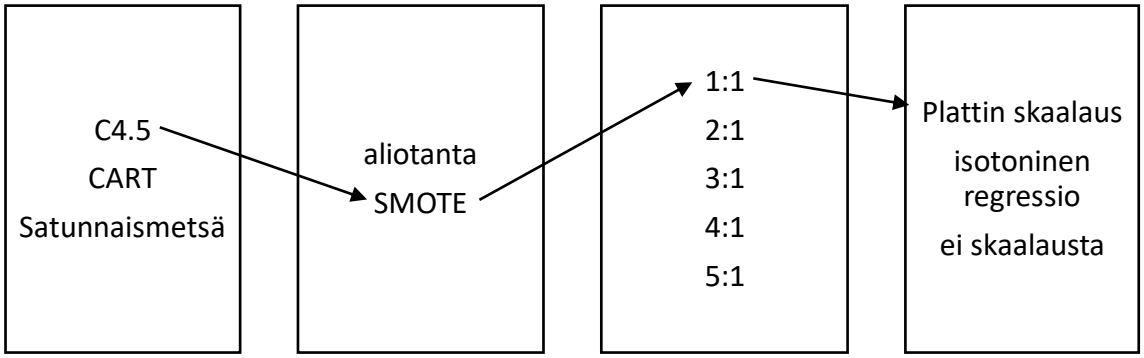
Mallien testauksessa käytetään *k*-kertaista ristiinvalidointia yleistettävyyden parantamiseksi. *K*-kertainen ristiinvalidointi on yleisesti käytetty menetelmä, jossa aineisto jaetaan *k*:hon suunnilleen samankokoiseen osajoukkoon, ja malli koulutetaan yhteensä *k* kertaa siten, että jokainen osajoukko on vuorollaan testiaineisto ja loput osajoukot muodostavat harjoitusaineiston. Näin harjoitusdata ja testidata ovat jokaisella kierroksella erilaiset, ja mallit testataan mallille ennen näkemättömällä datalla.

4 Testiasetelmat

Tähän tutkielmaan on valittu luokkajakaumaltaan epätasapainoisen datan käsittelyyn soveltuvia menetelmiä yhdistettynä päätöspuualgoritmeihin C4.5 ja CART sekä satunnaismetsään. Otantamenetelmistä testataan käyttämällä satunnaista aliotantaa ja SMOTEa viidellä eri negatiivisten ja positiivisten luokkien kokojen suhteella (1:1, 2:1, 3:1, 4:1, 5:1). Mallin koulutuksen jälkeen ennusteet kalibroidaan vaihtoehtoisesti joko Plattin skaalauksella tai isotonisella regressiolla. Lisäksi mallit testataan ilman kalibrointia.

Kuvassa 18 on esitelty kaikki metodit, joista muodostetaan testattavat yhdistelmät siten, että kustakin laatikosta valitaan yksi metodi yhdistelmään. Yhteensä eri yhdistelmiä tulee $3 \cdot 2 \cdot 5 \cdot 3 = 90$ eli kaikki mahdolliset yhdistelmät. Kuvassa havainnollistettu nuolin

yhdistelmä, jossa oppimisalgoritmina on C4.5-päätöspuu, otantamenetelmänä SMOTE luokkien kokojen suhteella 1:1 ja jossa saadut ennusteet kalibroidaan Plattin skaalauksella. Lopputuloksena raportoidaan alkuperäisellä jakaumalla olevan testiaineiston ennusteet ja niitä arvioidaan logaritmisen tappion, Brierin pisteiden sekä sekaannusmatriisin avulla eri raja-arvoja käyttäen.

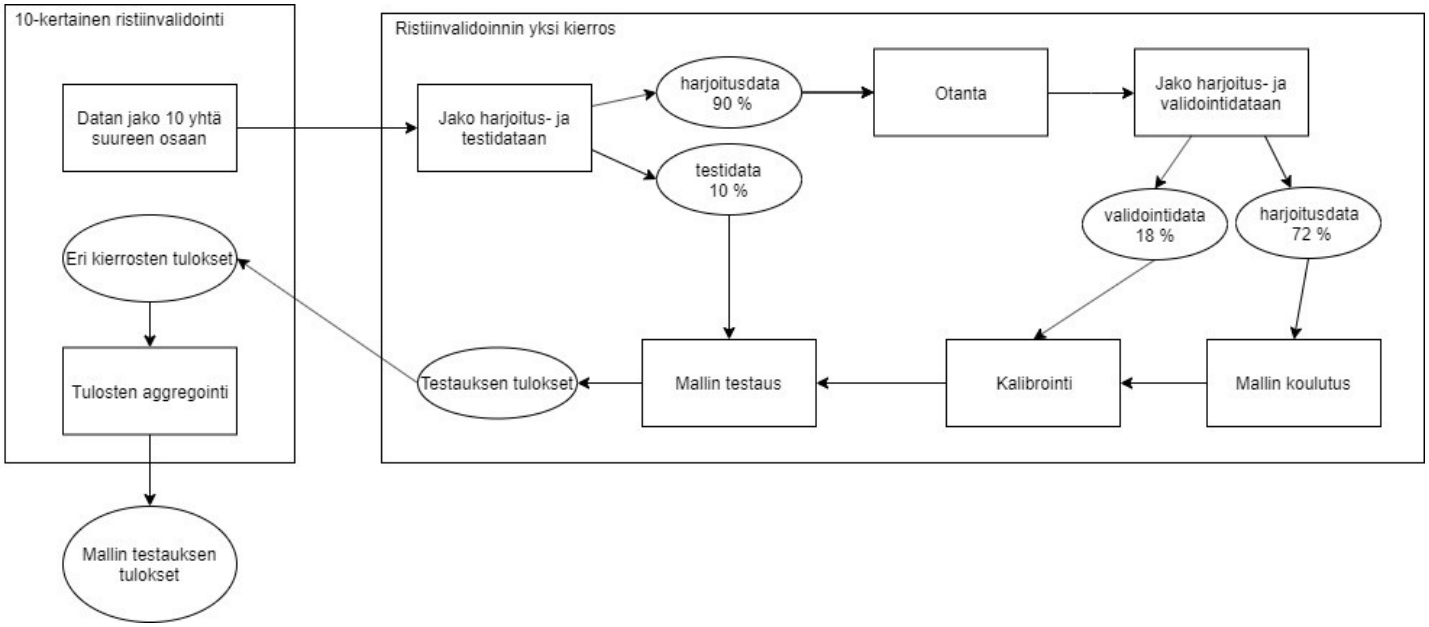


Kuva 18: Käytettävät metodit ja niistä muodostettavat yhdistelmät

Mallin harjoituksessa ja testauksessa hyödynnetään 10-kertaista ristiinvalidointia. Otanta ja kalibrointi tapahtuu ristiinvalidoinnin sisällä väärien tulkintojen välttämiseksi, joihin saatettaisiin päätyä, mikäli testidata ei vastaisi alkuperäistä luokkajakaumaa [Blagus and Lusa. 2015]. Kuvassa 19 esitetään ristiinvalidoinnin kierroksen kulku. Tämä toistetaan kunkin kuvassa 18 esitetyn yhdistelmän kohdalla.

Aluksi data jaetaan 10 kutakuinkin yhtä suureen osaan ristiinvalidointia varten. Kukin osa on kertaalleen testiaineisto, ja loput osat muodostavat sen kierroksen harjoitusaineiston. Vain harjoitusdataan tehdään otanta siten, että luokkien kokojen suhde on joko 1:1, 2:1, 3:1, 4:1 tai 5:1, jonka jälkeen tehdään jako varsinaiseen harjoitusaineistoon, jolla koulutetaan malli ja validointidataan, jota käytetään kalibrointifunktion muodostamiseen. Mallin koulutuksen ja ennusteiden (mahdollisen) kalibroinnin jälkeen mallia testataan alkuperäisellä jakaumalla olevalla testidatalla ja tämän testauksen tulokset raportoidaan.

Jokaisella ristiinvalidoinnin kierroksella tallennetaan testauksen tuloksena logaritminen tappio, Brierin pisteet sekä testidatan ennusteet. Lopullisena tuloksena mallin koulutuksen jälkeen raportoidaan logaritmisen tappion ja Brierin pisteiden keskiarvot sekä kootaan kaikki testidatan ennusteet yhteen jokaiselta kierrokselta sekaannusmatriisia varten. Koska testidata erotetaan ennen otannan tekemistä, testaamme nimenomaan mallin kykyä ennustaa tilaamisen todennäköisyyttä todellisen datan kanssa.



Kuva 19: Mallin opettamisprosessi

Toteutus on tehty R:llä ja hyödynnettävät funktiot ovat:

- C4.5: funktio *J48* paketissa *RWeka*
- CART: funktio *rpart1SE* paketissa *rpart*
- Satunnaismetsä: funktio *rf* paketissa *caret*
- Plattin skaalaus: funktion *glm* paketissa *stats* avulla (*family=binomial*)
- Isotoninen regressio: funktio *isoreg* paketissa *stats*

5 Tulokset

Tässä luvussa raportoidaan tulokset algoritmeittain ja vertaillaan tuloksia sekä logaritmisen tappion että Brierin pisteiden avulla. Taulukossa 4 esitellään päätöspuun C4.5 tuloksia eri otantamenetelmillä, otannan eri luokkasuhteilla sekä eri kalibrointimenetelmillä. Vastaavasti taulukossa 10 esitellään päätöspuun CART tulokset ja taulukossa 16 satunnaismetsän tulokset. Taulukoihin on korostettu pienimmät logaritmisen tappion sekä Brierin pisteiden arvot. Tämän lisäksi tutkitaan jokaisen algoritmin parasta versiota tarkemmin ja lasketaan sekaannusmatriisit muutamilla eri raja-arvoilla ja sekaannusmatriisia vastaavat evaluointimetriikat tarkkuus, sensitiivisyys, spesifisyys, luotettavuus ja F_1-pistearvo.

5.1 C4.5

Yleisesti ottaen SMOTE näyttäisi toimivan paremmin kuin aliotanta, ja mitä enemmän negatiivisia havaintoja, sitä parempi malli logaritmisella tappiolla ja Brierin pisteillä mitattuna. Aliotannan kanssa Plattin skaalaus parantaa tulosten logaritmista tappiota ja Brierin pisteitä ja kalibroi ennusteet paremmin kuin isotoninen regressio. SMOTEn kanssa isotoninen regressio puolestaan kalibroi ennusteet paremmin. Erikoisena poikkeuksena tässä on SMOTE luokkasuhteella 4:1 (taulukko 4).

C4.5-algoritmin kanssa parhaimpiin tuloksiin päästiin, kun käytettiin otantamenetelmänä SMOTEa luokkasuhteella 5:1 ja kalibroinnissa isotonista regressiota. Sekä logaritminen tappio että Brierin pisteet ovat pienimmät tällä yhdistelmällä.

Otantamenetelmä	Ei kalibrointia		Plattin skaalaus		Isotoninen regressio	
	logLoss	Brier	logLoss	Brier	logLoss	Brier
Aliotanta, 1:1	5,862	0,343	0,680	0,243	5,389	0,313
Aliotanta, 2:1	3,949	0,205	0,408	0,116	3,310	0,182
Aliotanta, 3:1	2,612	0,138	0,297	0,070	1,237	0,097
Aliotanta, 4:1	1,835	0,100	0,244	0,051	1,097	0,078
Aliotanta, 5:1	1,256	0,073	0,209	0,041	0,683	0,062
Aliotanta, ka	3,103	0,172	0,368	0,104	2,343	0,146
SMOTE, 1:1	4,431	0,253	0,605	0,207	0,230	0,031
SMOTE, 2:1	2,787	0,144	0,365	0,105	0,228	0,032
SMOTE, 3:1	1,871	0,098	0,267	0,069	0,191	0,023
SMOTE, 4:1	1,399	0,075	0,219	0,053	3,558	0,118
SMOTE, 5:1	1,089	0,059	0,189	0,043	**0,172**	**0,022**
SMOTE, ka	2,315	0,126	0,329	0,095	0,876	0,045

Taulukko 4: Päätöspuun C4.5 evaluointitulokset

Tutkitaan tarkemmin tämän mallin testiaineiston ennusteita sekaannusmatriisissa, kun numeeriset ennusteet luokitellaan positiiviseksi ja negatiiviseksi muutamalla eri raja-arvon t avulla. Tutkitaan eritoten, miten positiivisia havaintoja saadaan tunnistettua. Mallin ennusteiden jakauma on taulukossa 5. Maksimissaan tilaamisen todennäköisyys on 75 %.

Min	**25 %**	**50 %**	**75 %**	**Max**
0,000	0,002	0,006	0,025	0,750

Taulukko 5: Parhaan mallin (C4.5) ennusteiden jakauma

Alkuperäisessä datassa positiivisia havaintoja oli 1,8 %, joten testataan muutamaa raja-arvoa tämän ympäriltä. Raja-arvolla $t = 0{,}02$, saadaan taulukossa 6 oleva sekaannusmatriisi ja taulukossa 7 olevat tunnusluvut. Koska datan luokkajakauma on hyvin vino, tarkkuus ja spesifisyys ovat lähellä toisiaan. Kaikista negatiivisista havainnoista 76 % tunnistetaan negatiiviseksi ja kaikista positiivisista tunnistetaan 34 % positiiviseksi.

Liiketoiminnan kannalta on tärkeämpää tunnistaa positiivisia havaintoja. Väärät positiiviset eivät ole siksi niin haitallisia. Esimerkiksi kohdennustarkoituksissa on luultavasti parempi tunnistaa mahdollisimman moni potentiaalinen tilaaja sen kustannuksella, että kohdennusta osoitetaan myös heille, jotka eivät ole kiinnostuneet alkuunkaan tilaamisesta. Itse asiassa tärkeintä olisi saavuttaa kävijät, jotka ovat rajamailla: heille kohdentamisessa voisi olla eniten potentiaalia. Luokittelumallin väärät positiiviset ovat siis myös kiinnostava joukko, koska ne muistuttavat jollain tapaa tilanneita, vaikka eivät todellisuudessa olleetkaan tilanneita. Voisiko juuri heidät saada tilaamaan? Toisaalta, mikäli kävijä ei ole kiinnostunut tilaamisesta alkuunkaan, ei hänenkään käyttökokemustaan kannata häiritä turhalla kohdennuksella. Luultavasti kohdennukseen käytettävä tila sivustolla on kannattavampaa käyttää johonkin muuhun, esimerkiksi mainontaan.

t = 0,02		Todellinen	
		Pos.	Neg.
Ennuste	Pos.	2000	80068
	Neg.	3977	252017

Taulukko 6: Sekaannusmatriisi raja-arvolla *t* = 0,02 (C4.5)

Tarkkuus	0,751
Sensitiivisyys	0,335
Spesifisyys	0,759
Luotettavuus	0,024
F_1-pistearvo	0,045

Taulukko 7: Tunnusluvut raja-arvolla *t* = 0,02 (C4.5)

Raja-arvoa pienentämällä saadaan suurempi osuus positiivisia havaintoja oikein väärien negatiivisten kasvun kustannuksella. Raja-arvolla $t = 0{,}01$ saadaan taulukoissa 8 ja 9 annetut tulokset. Noin 47 % positiivisista havainnoista ennustetaan jo oikein, mutta vastaavasti negatiivisista havainnoista noin 40 % ennustetaan väärin.

t = 0,01		Todellinen	
		Pos.	Neg.
Ennuste	Pos.	2785	130516
	Neg.	3192	201569

Taulukko 8: Sekaannusmatriisi raja-arvolla t = 0,01 (C4.5)

Tarkkuus	0,605
Sensitiivisyys	0,466
Spesifisyys	0,607
Luotettavuus	0,021
F_1-pistearvo	0,040

Taulukko 9: Tunnusluvut raja-arvolla t = 0,01 (C4.5)

5.2 CART

Kuten algoritmin C4.5 kanssa, SMOTE näyttäisi toimivan yleisesti ottaen paremmin kuin aliotanta ja mitä enemmän negatiivisia havaintoja mukana, sen paremmin malli toimii sekä logaritmisella tappiolla että Brierin pisteillä mitattuna. Aliotannan kanssa Plattin skaalaus parantaa hiukan logaritmista tappiota verrattuna kalibroimattomiin tuloksiin. Sen sijaan SMOTEn kanssa isotoninen regressio kalibroi ennusteet paremmin (taulukko 10). Isotoninen regressio ei toimi yhdistettynä aliotantaan. Taulukon puuttuvat arvot johtuvat siitä, että kaikki validaatiodatan ennusteet ovat saaneet saman arvon eikä isotoniseen regressioon perustuva kalibrointi onnistu.

Logistisen tappion perusteella paras malli on SMOTE suhteella 5:1 ja yllättäen Plattin skaalauksella. Brierin pisteiden perusteella paras malli on, kuten algoritmilla C4.5, SMOTE suhteella 5:1 ja isotoninen regressio kalibrointitapana. Koska tällä mallilla on myös logaritminen tappio lähes yhtä pieni kuin parhaalla mallilla logaritmisen tappion perusteella, tutkitaan parhaat Brierin pisteet saanutta mallia tarkemmin sekaannusmatriisin ja muutaman eri raja-arvon avulla.

Otantamenetelmä	Ei kalibrointia		Plattin skaalaus		Isotoninen regressio	
	logLoss	Brier	logLoss	Brier	logLoss	Brier
Aliotanta, 1:1	0,665	0,237	0,667	0,238	7,185	0,417
Aliotanta, 2:1	0,414	0,116	0,419	0,118	-	-
Aliotanta, 3:1	0,303	0,071	0,301	0,070	-	-
Aliotanta, 4:1	0,246	0,051	0,243	0,050	-	-
Aliotanta, 5:1	0,211	0,040	0,207	0,039	-	-
Aliotanta, ka	0,368	0,103	0,367	0,103	7,185	0,417
SMOTE, 1:1	0,627	0,219	0,640	0,225	0,448	0,138
SMOTE, 2:1	0,374	0,101	0,379	0,104	0,263	0,045
SMOTE, 3:1	0,278	0,063	0,280	0,064	0,225	0,028
SMOTE, 4:1	0,224	0,046	0,226	0,046	0,204	0,025
SMOTE, 5:1	0,193	0,037	**0,191**	0,036	0,194	**0,022**
SMOTE, ka	0,339	0,093	0,343	0,095	0,267	0,052

Taulukko 10: Päätöspuun CART evaluointitulokset

Taulukosta 11 nähdään, miten positiivisen luokan ennusteet jakautuvat, kun maksimitodennäköisyys on 67 %. Raja-arvolla $t = 0{,}02$ saatu sekaannusmatriisi on taulukossa 12 ja evaluointimetriikat taulukossa 13. Taulukosta 13 huomataan, että tällä raja-arvolla saadaan 86 % positiivisista havainnoista luokiteltua oikein, mutta vastaavasti vain 28 % negatiivisista havainnoista. Jotta saataisiin kokonaisuudessaan paremmat luokittelutulokset, kasvatetaan raja-arvoa t. Raja-arvolla $t = 0{,}08$ saadaan sekä sensitiivisyys ja spesifisyys noin 57 %:iin (taulukot 14 ja 15).

Min	**25 %**	**50 %**	**75 %**	**Max**
0,000	0,000	0,066	0,099	0,673

Taulukko 11: Parhaan mallin (CART) ennusteiden jakauma

t = 0,02		Todellinen	
		Pos.	Neg.
Ennuste	Pos.	5112	238390
	Neg.	865	93695

Taulukko 12: Sekaannusmatriisi raja-arvolla *t* = 0,02 (CART)

Tarkkuus	0,292
Sensitiivisyys	0,855
Spesifisyys	0,282
Luotettavuus	0,021
F_1-pistearvo	0,041

Taulukko 13: Tunnusluvut raja-arvolla t = 0,02 (CART)

t = 0,08		Todellinen	
		Pos.	Neg.
Ennuste	Pos.	3438	142168
	Neg.	2539	189917

Taulukko 14: Sekaannusmatriisi raja-arvolla t = 0,08 (CART)

Tarkkuus	0,572
Sensitiivisyys	0,575
Spesifisyys	0,572
Luotettavuus	0,024
F_1-pistearvo	0,041

Taulukko 15: Tunnusluvut raja-arvolla t = 0,08 (CART)

5.3 Satunnaismetsä

Taulukossa 16 nähdään satunnaismetsän tulokset eri otantamenetelmillä ja kalibroinneilla. Parhaimmat tulokset saadaan SMOTE:lla ja vinommilla luokkasuhteilla, kuten päätöspuillakin. Satunnaismetsän kohdalla suhde 4:1 on hiukan parempi verrattuna suhteeseen 5:1. Tässä aliotannan kanssa isotoninen regressio toimii hiukan paremmin kuin Plattin skaalaus, vaikka päätöspuiden kohdalla Plattin skaalaus kalibroi ennusteet paremmin. Isotonisella regressiolla saadaan kauttaaltaan parempia tuloksia verrattuna siihen, että kalibrointi jätetään tekemättä tai että ennusteet kalibroitaisiin Plattin skaalauksella.

Otantamenetelmä	Ei kalibrointia		Plattin skaalaus		Isotoninen regressio	
	logLoss	Brier	logLoss	Brier	logLoss	Brier
Aliotanta, 1:1	0,640	0,225	0,627	0,220	0,609	0,196
Aliotanta, 2:1	0,397	0,121	0,393	0,114	0,371	0,104
Aliotanta, 3:1	0,299	0,081	0,293	0,073	0,247	0,060
Aliotanta, 4:1	0,248	0,063	0,233	0,052	0,206	0,045
Aliotanta, 5:1	0,214	0,051	0,200	0,042	0,189	0,039
Aliotanta, ka	0,360	0,108	0,349	0,100	0,324	0,089
SMOTE, 1:1	0,465	0,149	0,451	0,145	0,131	0,025
SMOTE, 2:1	0,284	0,076	0,270	0,074	0,113	0,023
SMOTE, 3:1	0,218	0,052	0,205	0,051	0,108	0,021
SMOTE, 4:1	0,183	0,040	0,171	0,040	**0,100**	**0,020**
SMOTE, 5:1	0,164	0,034	0,149	0,034	0,105	**0,020**
SMOTE, ka	0,263	0,070	0,249	0,069	0,111	0,022

Taulukko 16: Satunnaismetsän evaluointitulokset

Tutkitaan sekaannusmatriisin ja sen evaluointimetriikoiden avulla parasta satunnaismetsällä tehtyä mallia, jossa otanta tehtiin SMOTE:lla luokkasuhteella 4:1 ja ennusteet kalibroitiin isotonisella regressiolla. Mallin ennusteiden jakauma nähdään taulukossa 17, sekaannusmatriisi taulukossa 18 ja sen evaluointimetriikat taulukossa 19. Taulukosta 19 huomataan, että raja-arvolla $t = 0{,}02$ sekä positiivisista että negatiivisista havainnoista tunnistetaan oikein 60 %.

Min	**25 %**	**50 %**	**75 %**	**Max**
0,000	0,006	0,015	0,028	1,000

Taulukko 17: Parhaan mallin (satunnaismetsä) ennusteiden jakauma

$t = 0{,}02$		Todellinen	
		Pos.	Neg.
Ennuste	Pos.	3586	134211
	Neg.	2391	197874

Taulukko 18: Sekaannusmatriisi raja-arvolla $t = 0{,}02$ (satunnaismetsä)

Tarkkuus	0,600
Sensitiivisyys	0,600
Spesifisyys	0,600
Luotettavuus	0,026
F_1-pistearvo	0,050

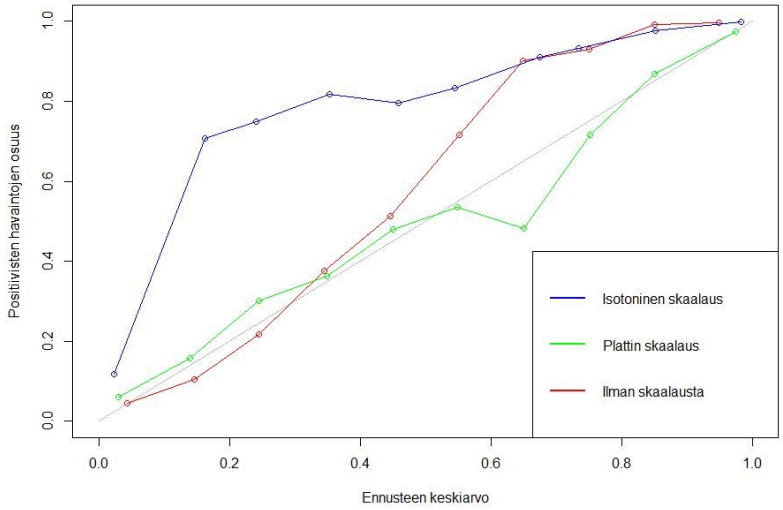
Taulukko 19: Tunnusluvut raja-arvolla $t = 0{,}02$ (satunnaismetsä)

Satunnaismetsän tulokset ovat jonkin verran paremmat verrattuna parhaaseen C4.5- tai CART-päätöspuihun. Satunnaismetsällä Brierin pisteet paranivat noin 9 % verrattuna molempiin parhaisiin päätöspuihin ja logaritminen tappio pieneni noin 42 % C4.5-päätöspuuhun verrattuna ja noin 48 % CART-päätöspuuhun verrattuna. Kaikilla valituilla algoritmeilla SMOTE yhdistettynä isotoniseen regressioon toimi parhaiten. Lisäksi parhaimmat tulokset saatiin, kun negatiivisten havaintojen määrää kasvatettiin harjoitusaineistossa.

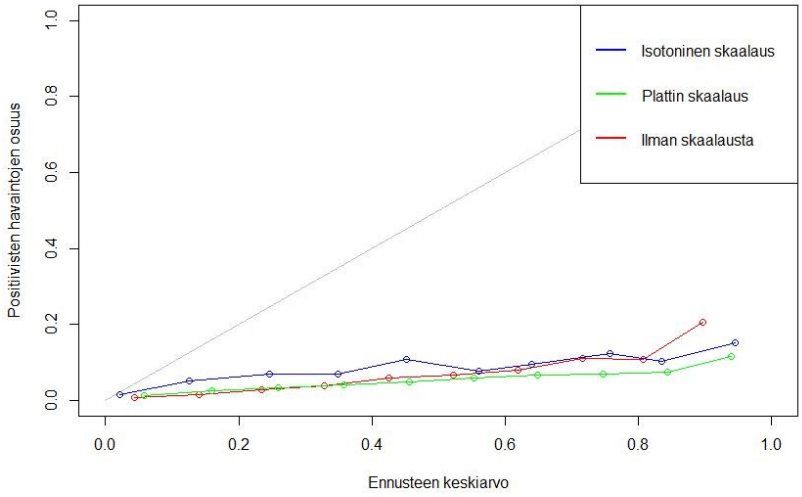
Mallin luokittelukyky jää melko heikoksi käytetyllä datalla ja valituilla menetelmillä. Mikäli ennusteista halutaan siirtyä luokittelumalliin niin, esimerkiksi raja-arvo $t = 0{,}02$ tunnistaa positiivisia havaintoja niin, että myös negatiivisten havaintojen erottelukyky säilyy kohtuullisena. Mikäli positiivisten havaintojen suurempaa tunnistamisprosenttia

pidetään tärkeänä, raja-arvoa t voidaan kuitenkin pienentää, esimerkiksi raja-arvolla $t = 0{,}01$ saadaan sensitiivisyydeksi jopa 0,79. Tällöin spesifisyys laskee arvoon 0,39 ja luotettavuus tippuu arvoon 0,023. Saamme siis tarvittaessa luokiteltua hyvällä prosentilla tilanneita kävijöitä tilanneiksi, mutta luotettavuus on heikkoa: tilanneiden joukossa on iso määrä oikeasti ei-tilanneita kävijöitä.

Jos kalibrointikuvio piirretään validaatiodatalla näyttäisi, että kalibrointi on onnistuneinta Plattin skaalauksella (kuva 20). Lisäksi tämän perusteella todennäköisyysennusteet näyttäisivät olevan varsin luotettavia. Sama mallin heikko luokittelukyky nähdään kuitenkin, kun piirretään kalibrointikuvio testidatalla (kuva 21). Tässä näkyy myös se, että isotoninen regressio toimi hieman paremmin kuin Plattin skaalaus.

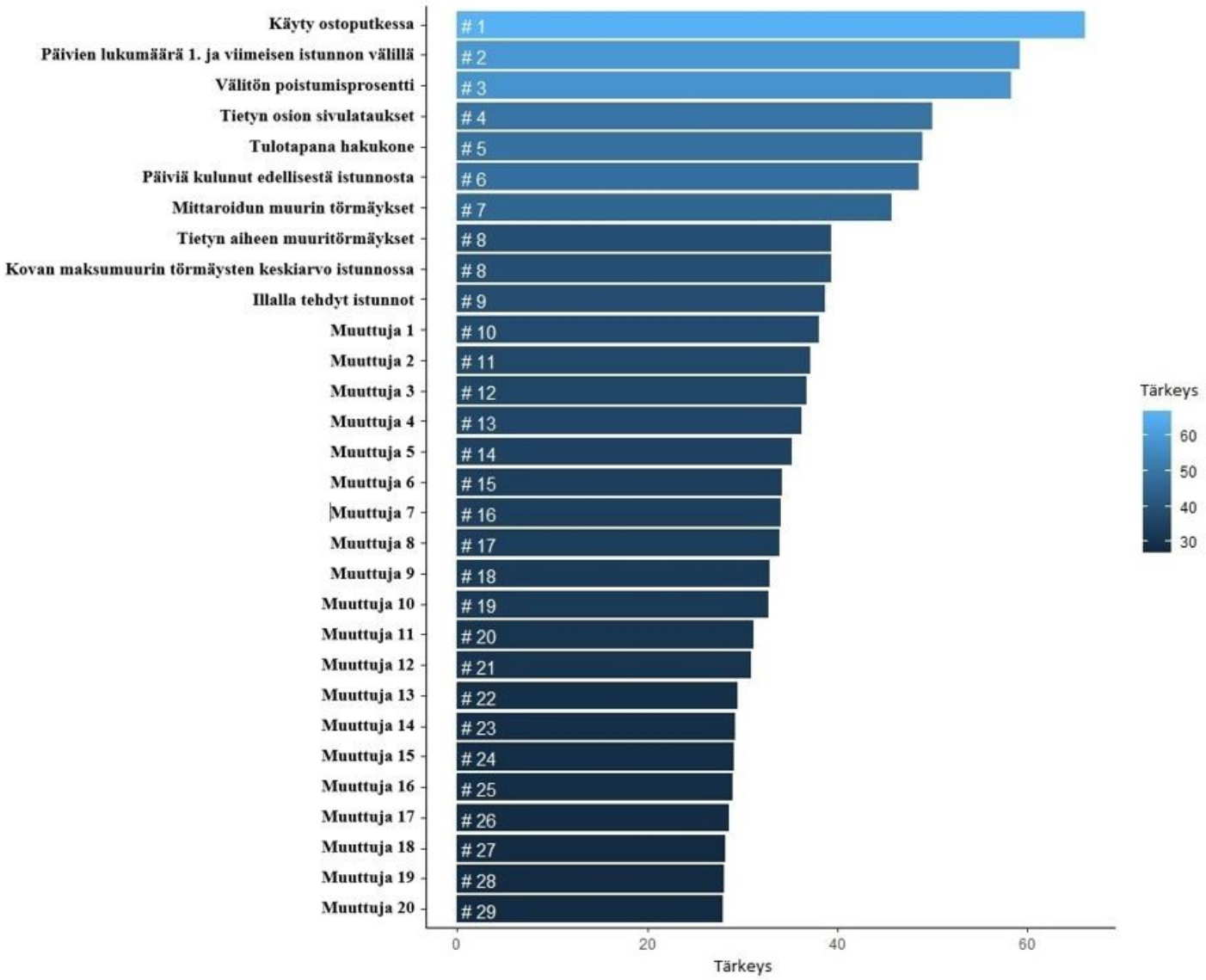


Kuva 20: Parhaan mallin validaatiodatan kalibrointikuvio



Kuva 21: Parhaan mallin testidatan kalibrointikuvio

Parhaan mallin 30 tärkeintä muuttujaa näkyvät kuvassa 22. Muuttujan tärkeydellä tarkoitetaan keskivähenemää tarkkuudessa eli MDA:ta, joka esiteltiin luvussa 3.2.4. Mitä suurempi MDA:n arvo muuttujalla on, sitä tärkeämpi tekijä se on mallissa. Ostoputkessa käynti on selvästi tärkein tekijä. Toiseksi ja kolmanneksi tärkeimmät ovat päivien lukumäärä ensimmäisen ja viimeisen istunnon välillä ja välitön poistumisprosentti. Osa muuttujista on anonymisoitu.



Kuva 22: Parhaan mallin tärkeimmät muuttujat MDA:n perusteella

6 Yhteenveto ja johtopäätökset

Tutkielmassa tutkittiin, voiko verkkokäyttäytymisen perusteella ennustaa sivustolla vierailevan kävijän tilaamishalukkuutta ja tilaamistodennäköisyyttä. Todennäköisyyden ennustamiseen valittiin päätöspuualgoritmit C4.5 ja CART sekä päätöspuita yhdistävä satunnaismetsä, sillä ne ovat luonnostaan todennäköisyysperusteisia. Eri malleja logaritmisella tappiolla ja Brierin pisteillä evaluoitaessa satunnaismetsä osoittautui parhaaksi oppimisalgoritmiksi.

Sovellusongelmaan liittyvä data oli luokkamuuttujan suhteen epätasapainoinen, ja tilanneita kävijöitä oli vain 1,8 % kaikista havainnoista. Datan epätasapainon käsittelyssä hyödynnettiin otantamenetelmistä satunnaista aliotantaa sekä vähemmistöluokan havaintoja lisäävää SMOTEa eri luokkasuhteilla. Yleisesti SMOTE toimi satunnaista aliotantaa paremmin ja testatuista luokkasuhteista 5:1 ja 4:1 toimivat parhaiten. Yleisesti ottaen vaikuttaa, että tässä ongelmassa suurempi määrä negatiivisia havaintoja harjoitusdatassa paransi tuloksia, joten samoilla algoritmeilla voisi testata vielä vinompia luokkasuhteita.

Koska otanta muuttaa luokkasuhteita, se vaikuttaa myös ennustettuihin todennäköisyyksiin. Tämän vuoksi ennusteet kalibroitiin. Kalibrointimenetelmistä isotoninen regressio toimi paremmin kuin Plattin skaalaus, kun otantaa tehtiin SMOTElla. Kun otanta tehtiin satunnaisella aliotannalla, Plattin skaalaus toimi paremmin päätöspuiden kanssa, mutta satunnaismetsän kanssa isotoninen regressio toimi puolestaan jonkin verran paremmin.

Parhaan mallin (satunnaismetsä SMOTElla suhteessa 4:1 ja isotonisella regressiolla) toimivuuden tarkastelussa hyödynnettiin luokittelutehtävistä tuttua sekaannusmatriisia testaamalla mallin luokittelukykyä muutamilla eri raja-arvoilla. Testauksesta huomattiin, että mallin luokittelukyky jää melko heikoksi. Esimerkiksi raja-arvolla $t = 0{,}02$, saatiin noin 60 % positiivista ja negatiivisista havainnoista tunnistettua. Rajaa pienentämällä saadaan suurempi osa positiivista havainnoista tunnistettua, mikäli suurentunut määrä vääriä positiivisia ei ole ongelmallista.

Tulosten parantamiseksi voisi testata lisää potentiaalisia puupohjaisia algoritmeja, kuten AdaBoostia [Freund and Schapire. 1997] tai XGBoostia [Chen and Guestrin. 2016]. Nämä ovat toimineet monissa yhteyksissä hyvin ja parantaneet tuloksia muihin algoritmeihin verrattuna. Datan laatuakin voi yrittää parantaa keräämällä ja käyttämällä parempaa henkilön tunnistetietoa saataisiin laadukkaampaa dataa, mutta toisaalta sitä

olisi luultavasti huomattavasti vähemmän. Myös uudet muuttujat voisivat auttaa mallintamisessa.

Lisäksi sovellusongelman luonne huomioiden, ja erityisesti huomio väärien negatiivisten tärkeydestä, voisi myös toisenlainen lähestymistapa kannattaa. Luokittelun sijaan voisi toimia niin kutsuttu kaksoisolentomallinnus (*engl. look-alike modeling, LAM*), koska sovellusongelma on varsin samantyyppinen kuin mainonnan kohdentaminen, jossa kaksoisolentomallinnusta on kokeiltu [Mangalampalli, et al. 2011]. Kaksoisolento-mallinnuksella mallinnettaisiin tilaajien käytöstä ja etsittäisiin heitä muistuttavia kävijöitä. Tämä ryhmä olisi tilaajakannan kasvattamisen kannalta oleellinen joukko, jolle voisi esimerkiksi kohdentaa markkinointia.

7 Viiteluettelo

BLAGUS, R. AND LUSA, L. 2015. Joint use of over- and under-sampling techniques and cross-validation for the development and assessment of prediction models. *BMC Bioinformatics* 16, 363.

BOULESTEIX, A., BENDER, A., LORENZO BERMEJO, J. AND STROBL, C. 2011. Random forest Gini importance favours SNPs with large minor allele frequency: Impact, sources and recommendations. *Briefings in Bioinformatics* 13, 292-304.

BREIMAN, L. 1996. Technical note: Some properties of splitting criteria. *Machine Learning* 24, 41-47.

BREIMAN, L. 2001. Random forests. *Machine Learning* 45, 1, 5-32.

BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A. AND STONE, C.J. 1984. *Classification and Regression Trees*. Wadsworth, Belmont, California.

BRIER, G.W. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78, 1-3.

BRÖCKER, J. AND SMITH, L.A. 2007. Increasing the reliability of reliability diagrams. *Weather and Forecasting* 22, 651-661.

CHAWLA, N.V., BOWYER, K.W., HALL, L.O. AND KEGELMEYER, W.P. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321-357.

CHEN, T. AND GUESTRIN C. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16).* ACM, New York, NY, USA, 785-794.

DAL POZZOLO, A., CAELEN, O. AND BONTEMPI, G. 2015. When is undersampling effective in unbalanced classification tasks? APPICE A., RODRIGUES P., SANTOS COSTA V., SOARES C., GAMA J., JORGE A. (eds) In *Machine Learning and Knowledge Discovery in Databases.* ECML PKDD 2015. *Lecture Notes in Computer Science, vol 9284. Springer, Cham,* 200-215.

Deep Learning Course Wiki. 2017. Log Loss. Retrieved from *http://wiki.fast.ai/index.php/Log_Loss*. Accessed 1.12.2019.

ELKAN, C. AND ZADROZNY, B. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayes classifiers. *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 609-616.

ELKAN, C. AND ZADROZNY, B. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 694-699.

FREUND, Y. AND SCHAPIRE R.E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55, 1, 119-139.

HE, H. AND GARCIA, E.A. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 1263-1284.

HAN, J. AND KAMBER, M. 2001. *Data mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco.

HO, T.K. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8, 832-844.

HSSINA, B., MERBOUHA, A., EZZIKOURI, H. AND ERRITALI, M. 2014. A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications* 4, 2, 13-19.

JANITZA, S., TUTZ, G. AND BOULESTEIX, A. 2016. Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics & Data Analysis* 96, 57-73.

KOTSIANTIS, S., KANELLOPOULOS, D. AND PINTELAS, P. 2006. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering* 30, 25-36.

LEWIS, R.J. 2000. An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California.*

LIN, H., LIN, C. AND WENG, R.C. 2007. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning* 68, 267-276.

LIN, W., TSAI, C., HU, Y. AND JHANG, J. 2017. Clustering-based undersampling in class-imbalanced data. *Information Sciences* 409-410, 17-26.

MANGALAMPALLI, A., RATNAPARKHI, A., HATCH, A.O., BAGHERJEIRAN, A., PAREKH, R. AND PUDI, V. 2011. A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns. *In Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)*. ACM, New York, NY, USA, 85-86.

NICULESCU-MIZIL, A. AND CARUANA, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning (ICML '05).* ACM, New York, NY, USA, 625-632.

PLATT, J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers* 10, 3, 61-74.

PRESS, W.H., TEUKOLSKY, S.A., VETTERLING, W.T. AND FLANNERY, B.P. 1992. *Numerical Recipes in C: The Art of Scientific Computing (2nd. ed.).* Cambridge University Press, Cambridge.

QUINLAN, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

ROBERTSON, T., WRIGHT, F. AND DYKSTRA, R. 1988. *Order Restricted Statistical Inference*. John Wiley and Sons, New York.

RUGGIERI, S. 2002. Efficient C4.5 [classification algorithm]. *IEEE Transactions on Knowledge and Data Engineering* 14, 2, 438-444.

STROBL, C., BOULESTEIX, A., ZEILEIS, A. AND HOTHORN, T. 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8, 25.

STEINBERG, D. AND COLLA, P. CART: *Tree-Structured Non-Parametric Data Analysis*. San Diego, CA: Salford Systems, 1995.

VISA, S. AND RALESCU, A. 2005. Issues in mining imbalanced data sets - A review paper. In *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference*. Dayton, Ohio, USA, 67-73.

YOHANNES, Y. AND WEBB, P. 1999. Classification and Regression Trees, CART: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity. Microcomputers in Policy Research 3, Food Policy Research Institute, Washington, D.C. USA.